
**Universitat de les Illes Balears**

# Ontology Matching based On Class Context: to solve interoperability problem at Semantic Web

Defended by

**Isaac Lera**

A thesis submitted to *Departament de Ciències Matemàtiques i Informàtica* of the University of Balearic Islands in accordance with the requirements for the degree of

**Doctor of Computer Science**

Thesis Advisor

Dr. Carlos Juiz

2012

# Acknowledgements

Este trabajo hubiera sido prácticamente imposible sin los pilares que sujetan mi cabeza y mi vida:

Agradecer a Carlos Juiz por su apoyo y aguante en todos los momentos buenos y malos, por darme la flexibilidad y la libertad para crecer como persona e investigador y, en definitiva, por proporcionarme *retos* y un modelo a seguir.

Donar les gràcies a Ramon Puigjaner per donar-me suport i acolliment formant part d'aquesta universitat.

Thanks to Nigel Thomas to hosting me at University of Newcastle Upon Tyne. He was an excellent host.

Agradecer a todos los que pasaron, están y seguirán estando por el *lab*, nuestro meeting point, con sus sugerencias, puntos de vista y por conversaciones interesantes sobre otros aspectos no relacionados con la tesis: Carlos Guerrero, Mehdi Khouja, Jaume Vicens, Pere Pau, Diana Arellano, Xavi Varona, y muchos otros a los que no nombro pero de vez en cuando pasan y alegran nuestros momentos.

Quiero agradecer a mi padres, Julián y Fidela, y a mi hermana Begoña el tiempo que me han dedicado.

Por último, agradecer a la persona que me enseño a ser consciente de los pilares y me ha soportado en esta travesía, gràcies Neus!

# Preface

When we look at the amount of resources to convert formats to other formats, that is to say, to make information systems useful, it is the time when we realise that our communication model is inefficient. The transformation of information, as well as the transformation of energy, remains inefficient for the efficiency of the converters. In this work, we propose a new way to "convert" information, we propose a mapping algorithm of semantic information based on the context of the information in order to redefine the framework where this paradigm merges with multiple techniques. Our main goal is to offer a new view where we can make further progress and, ultimately, streamline and minimize the communication chain in integration process.

## Resumen

Cuando observamos la cantidad de recursos destinados a convertir formatos en otros formatos, es decir transmitir una idea convirtiéndola útil para otra herramienta, es cuando comprendemos que el modelo de comunicación es ineficiente. La transformación de información, al igual que la transformación de la energía, sigue siendo ineficiente por la eficiencia de los convertidores. En esta tesis proponemos una nueva forma de "convertir" información, proponemos un algoritmo de mapeado de información semántica basado en el contexto de la información con el objetivo de redefinir el marco por donde este paradigma confluye con múltiples técnicas. Nuestro único objetivo es, por tanto, ofrecer una nueva visión por la cual realizar futuros progresos y, en definitiva, hacer más eficiente la cadena de comunicación facilitando la integración de información.

# Contents

# List of Figures

# 1

# Introduction

Interoperability is defined as the ability of two or more systems to exchange and use information.

From a natural point of view, we unaware of doing process of "interoperability" while we are talking, chatting, seeing or writing. Interoperability is to discover with a certain probability the constructor of a thing which is in our mind with the constructor feeling, transmitted, or observed [17; 18]. A constructor is the ideal object that defines one thing, when I write the concept *tree* the reader can think in *green leafs*, *branches*, *trunk*, and *roots*. *Tree* word triggers an explosion of words and relationships among then in our minds. Of course, each tree depends upon a subject's mind. My thought about a tree is one's having a lightly rounded top. Thus, the constructor is set by concepts, propositions, and proposition bodies (theories). Heterogeneity problem happens because our perception of the constructor is defined by a process of interpretation for providing meaning and obviously, when we want to communicate, ie, to change data with people.

## 1.1 About the problem

Data interoperability is not a new problem in computer systems. From a basic starting point, collaborative systems are more powerful than an isolate system. The exchanging information among systems takes place in all system layers: data, application and business, that it permits a collaboration a long time. From simple data models as plain schema files to actual web services, from basic rules to complex restrictions, and from a familiar advertisement to aggressive laws, data interpretation is essential to avoid unexpected and incoherent results.

Concerning the interoperability problem, the modelling language restricts available measures to do data comparison, a basic task of the process. The kind of modelling language has its own nature in terms of expressiveness, formality, utilization, prediction, ambiguity degree among others factors. The

discipline that encompasses these techniques, related to the comparison of data, is called mapping or matching. It receives other names according with the nature of its functions. There is other discipline more wide in a conceptual way than mapping it is called Schema Matching. Schema Matching is the task of matching between concepts describing the meaning of data in various heterogeneous and distributed data sources [47; 126]. Indistinctly, both ontology mapping and schema matching terms are used along this documents.

We have focused our approach in the paradigm of the Semantic Web (SW) since web languages have a huge influence in the communications around the world. SW is the idea of transforming of whole web data in information understandable for humans and machines. SW languages define unambiguous constructs to represent data which is easily handled through web operations enabling automated agents to access the web more intelligently. In other words, SW makes more efficient Knowledge Management (searching, extracting, maintaining, uncovering, and visualization information) and therefore, SW sets more measures that facilitate comparisons of data in interoperability problems.

SW languages are used to communicate things through an ontology model. An ontology provides constructors and a vocabulary which describe a domain in terms of axiomatic theories. Thus, the fact of using ontologies has been raised to a higher level of abstraction. In order to appreciate SW paradigm in our information society more detailed aspects are explained in the next sections and chapters. For that reason, this thesis is based on SW principles with the goal of proposing a new approach in this discipline called Ontology Matching.

### 1.1.1 Types of heterogeneity

Multiples causes generate information heterogeneity and most of them are inevitable. As aforementioned, ontologies and other type of languages suffer data heterogeneity interpretation. The causes of heterogeneity can be classified in four groups:

- Syntactic heterogeneity happens when two or more models of representation use different vocabularies, i.e.: language translations (English-Spanish,English-Danish,etc.), XML serialization and SQL, etc. It is solved defining equivalent or similar constructors between languages in a supervised way, i.e. English.*car* $\cong$ Spanish.*coche*, OWL.Class $\cong$ SQL.Table.
- Terminological heterogeneity occurs when we use different names for referring to the same entity, i.e.: *paper-article*, *book-volume*, etc.
- Conceptual heterogeneity happens due to the use of different axioms to define concepts or the use of different concepts to describe the same domain. Jones *et al.* provides a precise classification of these discrepancies [71]:
  - Different coverage (*Coverage*) happens when two models describe different regions of the domain with the same level of detail and from

the same perspective. For example, two domains as coal mine and gold mine share common concepts such as: mine concept description and some phases of the extraction process. At the same time, they differ in parts of the domain as part of the procurement process, product, etc. There is an overlap of the same term as the mine and its functionality.
  - Granularity difference happens when two models describe the same region of the world from the same perspective but with different levels of detail. For example, we could describe a forest as natural vegetation that grows in temperate regions on Earth, where there is different types of trees, or it can be distinguished natural deciduous forest vegetation dominated by trees hard as oak, beech, elm, linden, and maple, in the maritime climate is characteristic of evergreen holly in boreal, and so on.
  - Different perspective happens when two models describe the same region with the same level of detail, but from a different perspective. Using the example of the mine, a model can describe the stages of refinement of a product and other can describe the environmental impact.
- Semiotic heterogeneity is due to different interpretations of the same concept. Every person is different therefore it is also different the personality, temperament, beliefs, education or social network. This type of heterogeneity is quite difficult to solve if the concepts are named differently which is not typical. The appropriate choice of terms to represent the concepts is fundamental for a good design, as well as the correct definition of relations and functions with other elements. For example, if a person believes that places where selling costume jewellery can be considered as jewellery store then it makes an inappropriate interpretation comparing with a real jewellery store.

Conceptual and terminological differences are often more prevalent because they depend on domain knowledge and presentation functionality. And semiotic terminology differences are due to a bad use of vocabulary and a subjectivity of interpretation of the world, respectively.

### 1.1.2 The role of the Semantic Web

World Wide Web can be viewed as a set of interrelated documents which provide us more and more useful data. However, computers can hardly handle this information mainly for once reason: machines interchange data but do not understand the meaning of the data represented in the millions of linked documents. That problem, among others, was the seed that origins the idea of the Semantic Web by Sir Timothy Berners-Lee director and founder of World Wide Web Consortium [4].

Figure 1.1 represents all areas necessaries to “achieve” the SW goals: conceptual spaces of knowledge, automated tools, query answering, defining visible parts of information, web support, and so on. Detailed information of each

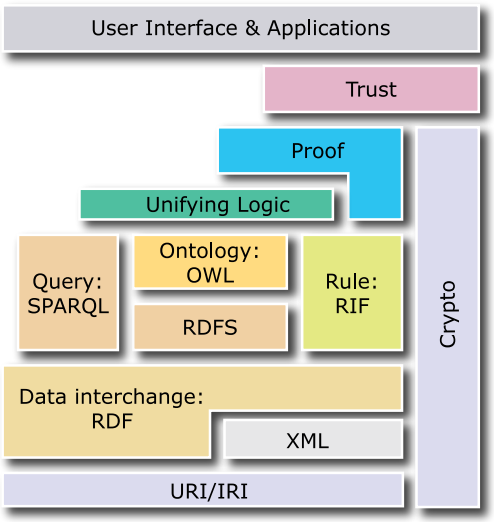


Fig. 1.1: Semantic Web Architecture by W3C

layer can be found in the next references [4; 156]. For the sake of clarity, we explain the layers regarding with the modelling language: XML, RDF, RDF schema, and OWL. XML is the syntax of the upper languages. It is based on nested and closed tags [158]. RDF language defines statements based on the structure of a triplet: subject, predicate, and object [157]. RDF schema defines the structure of RDF data. RDFs is considered the first semantic language of SW where there is a difference conceptual between the schema and facts. OWL constructors expand the interpretation and the logical capabilities of RDFs. Among these new constructors, we can reference some as: transitive, functional and symmetric properties, definition of classes by restrictions of other classes, existential and universal quantifiers, and cardinality. Basically, OWL is a language more expressive that previous one, but at the same time, it makes more difficult good designs of schemas and it decreases the performance of necessary reasoners to manipulate data.

RDF constructors and posterior languages follow a formal semantic theory which relates expressions to interpretations. "The following definition of an interpretation is couched in mathematical language, but what it amounts to intuitively is that an interpretation provides just enough information about a possible way the world might be - a 'possible world' - in order to fix the truth-value (true or false) of any ground RDF triple."[1]. It is only information to fix the truth-value of any ground triple. That fact causes that we know the meaning of this constructor and its relationships but we unknown the meaning contented in RDF triple. Although the content is less ambiguous still

[1] http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

it is. For example, our constructor of *elephant* contains the property "hasLeg" and also our constructor of *dog* among others characteristics. In OWL, we can define the class *elephant* as: $elephant = animal \sqcap hasLeg.exactly.4$ but at the same time, other designer can use the same axiom to define *dog* term: $dog = animal \sqcap hasLeg.exactly.4$. Of course, both definitions use OWL constructors, they are formally well defined but the meaning is not clear or it is not enough when we try to compare *dog* and *elephant* classes.

In any way, this kind of data communication requires also of new methods for managing data: address, storage, population, combination, interrogation, exportation, security polices, etc. Intrinsically, all these fields need other traditional disciplines such as: performance, software engineering, security, and so on. Furthermore, we can ask methodological questions: how can tools and techniques best be applied? and in which order? What about issues of quality control and resource management?... These relationships have set up a new field called Ontology Engineering [4].

Although SW languages decrease the ambiguity of the representation, it is still there. Both external operations (data integration, service orchestration or discovery, etc.) and internal operations (representation versioning, integration or querying, etc.) need to use mapping strategies to find out similarity in multiples data sources to combine them, doing a better web of linked data [155]. Dependencies among ontology engineering tasks and ontology matching (*ontology matcher*) tasks are drawn in the figure 1.2.

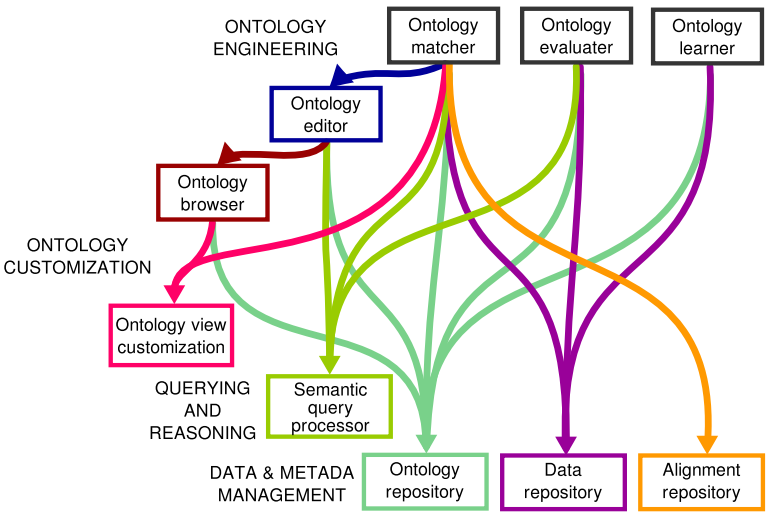


Fig. 1.2: Ontology Engineering parts and mutual dependencies. Figure by Neon Project, $6^{th}$ European Framework Programme.

### 1.1.3 Terminology

In order to clarify the suitable name of this area, according with [22; 37], there are related terms which different meaning to qualify some particularly part of the process, its functionality and mutual overlaps:

- **Matching** is the process to discover relationships between elements of different ontologies.
- **Alignment** is the output of matching. It is a set of correspondences.
- **Mapping**: An ontology mapping represents a function between alignments. Original ontologies are not changed, the mapping axioms describe how to express ontology elements in terms of the other ontology. Mainly, mappings are used for querying of different ontologies.
- **Combining**: Both ontologies are joined for a specific task where no information on how the relation is established can be given.
- **Integration**: One or more ontologies are used to define a new one. Two basic approaches are union and intersection.
- **Mediation**: Through an upper ontology, the process try to achieve interoperation between both sources reconciling their possibles differences.
- **Merging**: A new ontology is created from more ontologies where elements will be unified and replaced. Often we can not tracked back to their source.

The term mapping is used throughout this document, being the matching term equivalent in all purposes. One of our goals in this thesis is to catalyse all the processes in one since all share common and usable synergies. If we merge two ontologies or we create a new one it depends on our final goal but the process is the same. From our point of view, alignments are the results of a mapping system. An alignment is a logical relation between two elements; multiples relationships can be established (combining different sources); and there are new axioms to describe elements in function of external data.

### 1.1.4 Ontology Mapping

Mapping process tries to discover information about the closeness of a couple of concepts in function of the meaning, capabilities, features, among other characteristics. Mapping process has been widely researched since the advent of computer languages. With the apparition of new languages or models of representation, mapping strategies have been adapted to support them. Each new representation language increases the representation richness, which also improves the accuracy of mapping process, and extends research scientific borders. Nowadays, ontologies are the last paradigm by one of capacities of inferring new data due to are based on logic model.

An Ontology Mapping (OM) process tries to discover the similarity degree or the semantic relatedness of two elements of different ontologies. A mix of several algorithms, using all kind of information (from structural data, axioms, external resources as: catalogues, dictionaries, etc.), take part in this

process to calculate a value probability between 0 and 1. A degree of 1 indicates that both concepts are semantically equivalents, which it implies that they could be replaced in both contexts without problems of data incoherency. The rest of values degrade this relationship of equivalence increasing data incoherency. For example, concepts as *door*, *gate* and *portal* could be exchange in most of the contexts. Less obvious, will be exchange *door* for *windows*, and less, *door* for *stone*. Most of the algorithms or frameworks have as output this probability according internal criteria. Results are checked to guarantee a secure replacement. This human or machine verification is based on suppositions according to a numerical value which does not offer information about replacement context. A replacement context is defined as all suitable information that is essential to calculate the probability value. Thus, two concepts as *door* and *gate* are not equivalent in contexts as *flat interior design* and *cultural castle routes* respectively. We lose information that can cause sequences of failures in predictions.

Multiple mapping techniques are based on one to one comparison between ontology elements. Classifications of matching techniques are presented in the works of [41; 126]. These comparisons are based on three types of resemblance: labels, taxonomy structure or axioms using external resources as thesaurus or encyclopedias. Most of the cases, the outcomes are represented through a similarity probability (e.g. a car = an automobile with 95%)[40; 140]. The resemblances based on labels are essentials since elements with the same name are equals. Thus, methods based on morphological analysis of terms studies the behaviour and combination of morphemes. For instance, the word *unbreakable* has three morphemes: *un-* (meaning not x), a bound morpheme; *-break-*, a free morpheme; and *-able*, a free morpheme; *un-* is also a prefix, *-able* is a suffix. Other method is based on formulas to find the distance among names (e.g Hamming distance, Levensthein distance, Needleman-Wunch distance, etc.). Structural resemblances are based on element position in the taxonomy and on the number of relationships (as children or descendants, data types and objects properties). The use of external resources attempts to find out correspondences among terms with external knowledge.

### 1.1.5 Ontology Mapping Scenarios

Large number of applications or protocols have or have had some mapping models along their works: transformations among languages or layers architecture, services protocols, specific and internal data processing, etc. J. Euzenat *et al.* and M. Ehrig detail some basic scenarios [37; 41; 140]:

- Information Integration
- Peer-to-peer information sharing
- Web Service composition
- Autonomous communication systems
- Navigation and query answering on the web

## 1.2 Hypothesis

Multiples ontology mapping methodologies have been inspired from classical matching techniques adapting features to manage semantic particularities. Most of these types of approaches hardly use all potential of semantic representations, and normally they are isolate processes that do not combine different contributions. Good results are obtained but secondary aspects are ignored such as integration of results using same tools, unsupervised methods, or the loss of algorithm information created along the process.

Alignments represent a ratio of equivalence among entities by means of a confidence measurement, often a range of 0 and 1 ($[0, 1]$) using often lexical distributional similarity formulas. In other cases, it provides information on the type of relationship (equivalence or subsumption) that it represents a semantic relationship between both terms. In some cases, both data are combined.

In most of the approaches, authors do not explain the type of relationships that it is represented with a numeric value. Therefore, what is the semantic difference between a 0.75 and 0.7?, and what we should suppose about a numeric value of 0.8 in a subsumption relationship? A greater probability implies that it is suitable, better, more equivalent, more subsumed? Of course, the final application knows how to use the mapping information. That application defines, uses and interprets results according with its nature, but can we reuse the alignments represented with a numeric value? and can we interpret them?

From our point of view, the relation represented with a range of values is ambiguous in a semantic model. Let's assume the following model of interpretation. We analyse in detail the degree of confidence between 0 and 1 values and combinations. First assumption, generally it concerns a relationships of equivalence. Therefore, previous simple interval we can be split in the next intervals: [0], (0,1) and [1]. The value [1] defines the perfect match that can only occur when both classes have the same meaning. Independent of the context both are equals, where the meaning of each is determined by the semantic context. The interval (0,1) refers to the relationships of synonym. Two words are synonyms if both can be interchanged in their respective representations without producing changes in meaning. The last interval [0] refers to the relationship of antonym. It is here where it lies the problem of this type of representations. For example, two antonyms are `cold` and `heat`. Both words are antonyms but there is a word that relates: `temperature`. In this case, the interpretation is again influenced for the context. The existence of `temperature` word can cause that an algorithm rather provides a value above 0 than an absolute value of 0. There is no form to assure it: it has not been decided on a standard and there is no report on the matter. In an interval of [0,1] values, ratios of equivalence synonymy, and antonym can be assumed. The representation of an alignment as a pair of elements, and a numerical value between [0,1] is an ambiguous representation.

If we decide to choose the second case: the alignment is represented with a type of syntactic relation, regardless of the type of relationship chosen. The confidence degree is not useful to represent such relationship. That is to say, the type of relationship in the alignment through a logic constructor is the suitable thing. That constructor represents a semantic relationship, but the degree is something superfluous in semantic representations, only it is useful in computational tasks.

For this reason, we propose a representation that makes special emphasis in the correct standardization of the results. A part from this representation, there is some points to be mentioned: the role of meaning and the context, along with the relationships between them to determine the semantic type of relationship between two words.

Moreover, we explain basic steps to calculate lexical distributional similarity between two words in the next chapters. We explain how some of them have been applied, combined and adapted to the semantic representations in multiples approaches in the section of related work.

From our point of view, the baseline is located at intermediate situation where researchers apply traditional paradigms, based on lexical and structural analysis on semantic representations, instead of the use semantic information: interpretation, axioms or complex constructors. In this intermediate situation, we address our work at semantic level. Mapping algorithms need to work with the interpretation of the concept that it is defined by the context and its meaning.

We focus on mapping algorithms, specifically, on ontology mapping. The algorithm created in this word is called OMoCC, an algorithm of Ontology Mapping based On Class Context. OMoCC is only a tool which permits to show the necessity of a suitable semantic measure for mapping tasks, and besides of a standard language in semantic representation of alignments. We present an analysis of current measures and define as the minimum metric the use of the meaning and/or the context. If the algorithm deals with the context of the representation then it can manage any domain in unsupervised way. Moreover, we design an OWL-base language to define alignments and it promotes the usability of the alignments in multiples applications, and this language does not require specific libraries or applications to manage it.

OMoCC is the result of our research and tries to attain a series of objectives:

- We analyse the phases of ontology mapping process, and we will set out the similarity of the process with the methodology of the process of discovery of information in Information Retrieval paradigm.
- In the state-of-the-art, we propose a survey where we analyse and we describe each one of the metric and approaches to resolve OM problem.
- We analyse the most general metrics that exist in literature and we propose some facts to prove that the essential metric of semantic mapping is the management of the meaning and/or the context.

- Based on this metric, we propose some rules to discover the meaning of the elements of an ontology in an unsupervised way. Moreover, we try to synthesize the elements of an ontology more representative at the context of the representation using criteria to identify them.
- We represent alignments through a OWL-base language. It has the same functionality that original representations. Thus, the alignments are ontologies as well.

### 1.2.1 Hypothesis through an example

In order to clarify the forthcoming explanations, we use an example related to museums and art galleries, instead of the *library* example [37; 41]. Well known examples of private or public museums are: MOMA, Solomon R. Guggenheim Foundation, MNAC, Museo Nacional del Padro, Musée du Louvre, etc. These institutions promote the exchange of works and exhibitions among themselves to attract the public, to get some social and cultural impact, and to share culture with citizens. Each institution according with its enterprise policy manages different models of information. Some models are designed for specific purposes: audio guides, data visualization on web pages, internal cataloguing, and so on. To sum up, we could say that pieces of art are described according with global parameters, e.g., architectonic or drawing styles, and literally genres, with local or national information, e.g., authors' circumstances, military or politic conflicts, multiples owners, and restorations and with punctual thematic exhibitions, e.g., `Tim Burton Exhibition` (MOMA) and `Tesoros del Hermitage` (Museo del Padro).

Unfortunately, we do not have access to them but whether we have token a glance a MOMA and NY Guggenheim webs to extract enough data with the goal of providing illustrative and basic examples of OM cases[2] [3]. First example is related with the price of entrance tickets. Although there are three types of tickets, they do not contain the same information (see figure 1.3). In this little part of data, we can observe some common cases of heterogeneity. Thus, *Tickets* and *fees* are lexical different but we understand the meaning. *Adults* are equivalents. *Children tickets* and *children fees* are similar concepts where age restriction is different, a logical heterogeneity. In NY Guggenheim, Students and Senior older 65 have the same price. In contrast, Student idea does not appear in MOMA model. This example illustrate the complexity of taking decisions about the structure of the model and the relationships among parts of the structure. Alignments will depend on the interpretation of each person. In our opinion, some possible solutions are: $MOMA.Senior > 65$ is a subcategory of $NYGuggenheim.StudentsAndSenior > 65$, and $NYGuggenheim.Children < 12$ is a subcategory of $MOMA.Children < 16$ (blue lines).

[2] http://swap.uib.es/MOMA.owl
[3] http://swap.uib.es/NYGuggenheim.owl

We make a conceptual leap about the causes and the necessity of applying semantic matching on semantic representations and we explain both hypothesis in action. Regarding with the first one, we should think that we need to compare the `figure` element. Its interpretation depends on the context we can not assume the meaning without more related elements. If that was the case, `figure` could refer a mentionable person[4] or a short musical composition[5] Although, `figure` term appears in both representations or databases, we could make a serious trouble integrating musical compositions in a personal data store. Regarding with the second hypothesis, most of current algorithms could establish that between `picture` and `photography` there is a subsumption relation with a stronger degree of `0.823`. In function of this fact, which interpretation does this value have? Are there more relationships between both elements? author, work, piece of art,... It is indispensable to join both hypothesis since it is imprecise to discover new relationships without a suitable context. This happens in comparatives based on lexical similarities, for example, taking into account only lexical word formation where words such as `piece` and `pierce` have similar character formation but they have different meaning because they come from different contexts.

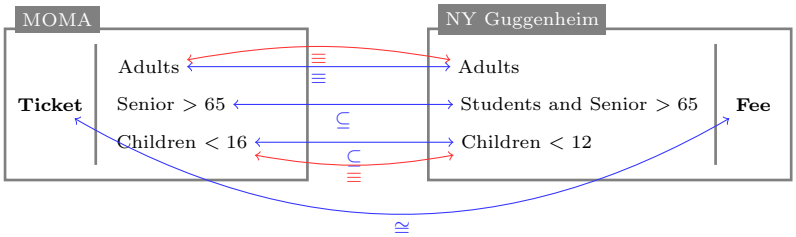


Fig. 1.3: Types of admission tickets according to MOMA and NY Guggenheim

[4] "a person of a particular kind, especially one who is important or distinctive in some way" Oxford Dictionary

[5] "short succession of notes producing a single impression; a brief melodic or rhythmic formula out of which longer passages are developed" Oxford Dictionary

## 1.3 Organization of this work

We present a chapter-by-chapter summary of the remainder of this thesis, which are classified in two parts:
*Part I. Ontology Matching: Background knowledge*

- *Chapter 2.* In this chapter, we clarify the real situation of this problem inside the diverse paradigms. Ontology mapping is regarding a basic task in other disciplines as Information Retrieval, Data Mining or Text Mining, but one notable difference the level of representation used.
- *Chapter 3.* This chapter is an extensive compilation of techniques applies on each phase of the general methodology. At the end, we discuss about the suitability of these measures.

*Part II. Contribution*

- *Chapter 4.* The main two branches of the contribution are introduced in this chapter. Each one is explained in detail in their respective chapters.
- *Chapter 5.* First detailed explanation is devoted to the role of the meaning and the context along the ontology mapping process.
- *Chapter 6.* This chapter regards with the language to represent results according a semantic paradigm. All constructors are explained and multiples examples permit clarify them.
- *Chapter 7.* Finally, both ideas are joined together in our algorithm called OMoCC.
- *Chapter 8.* These combinations of ideas are tested in a well-known benchmark. Because of certain functionalities of our work they have been possible to only realise partial tests of this benchmark.

Further explanations, cases of study, presentations of partial results and their corresponding analysis are included in annexes. Last *chapter 9* contains the thesis summary, the contributions, and the future work and applications.

To sum up, we comment on the wording in this official report where some chapters are written in Spanish. Thus, in order to unify content, the examples and figures appear in English. The use of acronyms in named of disciplines, techniques and names are also in English.

# Part I

## Ontology Matching: Background knowledge

# 2

# Disciplinas relacionadas y conceptos base

En este capítulo se asienta el conocimiento base para ubicar el mapeado de ontologías dentro de las ciencias de la computación. Presentamos dos enfoques. El primero de ellos es el filosófico: la definición de similitud de dos elementos como desencadenante del proceso de mapeado. El segundo enfoque es más práctico, el problema está orientado al campo de las ciencias. Desde nuestro punto de vista, el mapeado de ontologías es una técnica más dentro de las disciplinas que a continuación expondremos con la única salvedad distintiva: las fuentes de datos son representaciones semánticas de información.

Al ser elevado el número de disciplinas relacionadas, se ha simplificado la explicación de algunas de las áreas expuestas. Intentamos suplir con las pretensiones ávidas del lector mediante un número propicio de referencias.

## 2.1 Concepto de similitud

El mapeado de ontologías se define como un proceso que distingue el grado de similitud y el tipo de relación semántica que guardan dos conceptos de representaciones diferentes. El problema es bien conocido en la integración de datos: encontrar elementos comunes e integrarlos en diferentes fuentes y para ello, es necesario conocer hasta que punto son similares. Al nombrar esta otra disciplina podemos plantearnos una serie de cuestiones con el objetivo de ubicar el problema: ¿Cuáles son las disciplinas relacionadas al mapeado de ontologías? ¿Es la primera vez que aparece esta problemática? Si es así ¿podemos considerarlo único? Para iniciar esta andadura necesitamos responder a una cuestión más simple pero planteada antes de la aparición de los sistemas de computación. Con esta simple cuestión comprenderemos con objetividad las bases sobre las que se fundamentan las técnicas y la amplitud de este campo tan relativamente nuevo como es el mapeado de ontologías. La pregunta es: ¿qué es la similitud?

La principal ley que han de cumplir dos objetos similares, de la no formulada definición de similitud, fue definida por von Leibniz: “Eadem sunt

quorum unum potest substitui alteri salva veritate"[1]. Sin profundizar en el término de veracidad ni en el de equivalencia absoluta en el cálculo de la similitud en las representaciones semánticas vamos a suponer la existencia de un grado de similitud. Una teoría ajena a las equivalencias absolutas de von Leibniz pero sí adecuadas en términos de su contexto a la naturaleza de la representación. Simplemente no entramos a valorar las teorías respecto a la veracidad: ¿por qué hasta que punto el Dr. Jekyll y Mr. Hyde son la misma persona representando *seres* contrarios?

En términos lingüísticos, son los elementos de una representación los que nos interesa comparar. Los trabajos de Zellig S. Harris definen la hipótesis distribucional (*The Distributional Hypothesis*). Básicamente, la idea es que los elementos de una sentencia o una estructura están relacionados. Las palabras con propiedades distribucionales similares tienen significados parecidos. Harris defendía que era posible crear una topología de todo el lenguaje con respecto a un comportamiento distribucional. Estos primeros estudios establecen un punto de partida para determinar la similitud de las palabras, ya que permiten su comparación e intercambio en diferentes textos [116; 134]. Encontramos buenos ejemplos de este tipo de pensamiento donde el contexto y el significado están relacionados, [136] "words with similar meanings will occur with similar neighbors if enough text material is available", [133] "words which are similar in meaning occur in similar contexts", [46] "You shall know a word by the company it keeps", o los tres niveles de proximidad que define [52].

En esta área de investigación, hablamos de similitud léxica cuando todos los elementos de las representaciones tienen constructores léxicos. Son las representaciones más habituales en los sistemas computacionales. Dando lugar a una área de investigación llamada similitud léxica distribuida, donde dos palabras son distribucionalmente similares si aparecen en contextos similares.

Por otro lado, podríamos plantearnos la necesidad de ignorar la similitud léxica cuando trabajamos en modelos de representación semánticos. Sin embargo, en la similitud semántica la sustitución de dos palabras acorde al tipo de relación no ha de alterar la veracidad. Dos frases o representaciones pueden mantener la veracidad y tener diferente significado, pero dos frases con el mismo significado han de ser veraces. Por tanto, un requisito de la similitud semántica es la similitud léxica distribucional. Si pensamos en términos de niveles de lenguaje tal afirmación es compresible: "léxico $\supset$ sintáctico $\supset$ semántico". Tal como veremos en el siguiente capítulo, casi todas las propuestas en esta área dependen de trabajos relacionados o inspirados en el mapeado de léxico de conceptos -medidas de similitud léxica distribuida-.

En términos de similitud léxica, podemos sustituir un `animal` por un `perro` en ciertos contextos. En términos de similitud semántica, esta sustitución es posible pero la existencia de una relación semántica entre ambos complica la interpretación: ¿son equivalentes o uno subsume a otro? Si son equivalentes

[1] Las cosas son la misma cuando una puede ser sustituida por la otra sin perdida de verdad

`caballo` y `perro` son `animales`, entonces ¿`caballo` es sustituible por `perro`? Si son subsumidos ¿hasta qué punto se pueden sustituir en un contexto sin alterar la interpretación? La representación de los resultados es crucial para disminuir la ambigüedad de los mismos aunque la naturaleza de la aplicación asuma en cada caso su uso. Por esta causa, en esta tesis proponemos el desarrollo de un conjunto de constructores para representar los resultados de un algoritmo de mapeado de ontologías.

De la definición extraemos la operación indispensable de sustituir. Un elemento puede ser sustituido por dos enfoques uno puramente lógico o por la definición de la identidad [125]. El primer enfoque tiene un punto de vista más práctico, si la identidad tiene unas características lógicas que otra identidad también presenta, ambos pueden ser sustituidos sin perdidas de veracidad. El segundo, un enfoque más filosófico, el proceso de sustitución se sabe en la identificación de la identidad mediante su definición. Sin entrar en un debate entre ambas posturas, nosotros utilizamos el contexto y las propiedades lógicas de los elementos para averiguar el significado. A partir del significado de cada clase realizamos el proceso de equivalencia exacto o por similitud en etapas posteriores. Como no podía ser de otra manera el contexto desempeña un eje central en el descubrimiento de la definición exacta de la clase, tal como muestran por ejemplo los dos experimentos de [103].

## 2.2 El contexto de la comunicación

El contexto es el "entorno lingüístico del cual depende el sentido y el valor de una palabra, frase o fragmento considerados" [2]. El entorno lingüístico no tan sólo está influenciado por las relaciones semánticas entre las palabras, sino también por el nivel pragmático. El significado de una palabra o de una frase o de un documento o de cualquier modelo de representación depende del pragmatismo de la comunicación [146]. El nivel pragmático engloba factores relacionados con el grupo de personas implicados en la comunicación: su nivel intelectual, las relaciones que hay entre ellas, la personalidad, el carácter y la postura corporal, más otros factores como la cultura, la situación y lo que se quiere conseguir. Como ejemplo la palabra *fuego*, dependiendo de la entonación y del lugar podemos estar solicitando un encendedor o advertir de un incendio.

Podemos diferenciar dos perspectivas en la definición del contexto: las funcionales, desde el punto de vista de un sistema de información, y las cognitivas, el contexto como conocimiento. Por tanto, hay dos tipos de contextos: el contexto al conocimiento y el contexto respecto a una base de conocimientos o un sistema de información.

Las aproximaciones funcionales ofrecen un servicio en función del contexto situacional: ¿dónde está? ¿qué hace? ¿qué quiere?... eluden la definición for-

[2] fuente RAE

mal de contexto: ¿qué es? ¿hay alguna relación entre contextos? ¿de qué contexto proviene tal información?...

Uno de los primeros trabajos para representar el contexto desde un enfoque cognitivo se debe a [68], dentro de la Inteligencia Artificial (AI). Propone que los contextos son específicos a una conversación e introduce la idea de contextos generales respecto a otros, todo ello mediante el uso del lenguaje Prolog [24]. Le siguieron una serie de trabajos con el objetivo de clarificar la idea y representación del contexto en [69] y dos años después en [70]. En ellos se nombran nuevas relaciones entre contextos (por tiempo, por especialización y por *descontextualización*), el concepto de transcendencia en nuestros pensamientos y otras series de cuestiones relacionadas con el lenguaje y con los ejemplos propuestos.

Otro campo donde el contexto ha adquirido una especial importancia, tal vez derivados de las primeras investigaciones en AI, son los sistemas ubicuos [161]. Un sistema ubicuo es un sistema omnipresente proveedor de servicios personalizados. [149] realizan una síntesis de todos las propuestas para representar contextos conscientes (*awareness context*) mediante diferentes tipos de lenguajes. Ellos mismos desarrollan *Context Ontology Language* (CoOL) [150]. CoOL está definido bajo la sintaxis de OWL y permite representar situaciones absolutas de tiempo, duraciones, lugares geográficos, lugares simbólicos, secuencia de eventos, costes, distancias espaciales, condiciones climatológicas, aerolíneas y velocidades. La finalidad de la representación es la integración de servicios. En definitiva, es la pretensión de un sistema ubicuo, es decir, proporcionar los servicios más adecuados para cada usuario, en función de su viaje, ubicación, destino y momento.

En los sistemas ubicuos o en la gestión de servicios tal es la necesidad de acceder a los datos, a reutilizarlos y al uso de reglas de razonamiento que muchos de los trabajos actuales en este campo se basan en el uso de lenguajes semánticos [7].

Desde el punto de vista de la psicología cognitiva no se define la funcional del contexto (el lugar, las acciones, la necesidad, los costes, etc.), imposibilitando su manipulación y la interpretación del mensaje. Es [69] quien introduce la noción de operaciones sobre contextos respecto a las posibles relaciones. En trabajos recientes, [14] justifica la necesidad de formalizar el contexto del conocimiento para poder procesarlo y usarlo en aplicaciones sensibles al contexto.

En la propuesta de Kashyap y Sheth, publicada en 1996 [75], presentan la dualidad entre similitudes semánticas y de esquema en una base de datos. Línea continuada a lo largo de varias de sus publicaciones en común [138; 139]. Es significativo la creación del contexto a través de toda la información disponible del esquema y, como posteriormente, estas correspondencias son adaptadas al esquema en función de las restricciones definidas. Según los autores, "el contexto es el componente clave en la captura de la semántica relacionada en la definición de un objeto y sus relaciones con otros objetos". Citan múltiples trabajos relacionados con el mapeado de bases de datos de autores que sentaron la base en la área en comparativa de esquemas en la década de los

90, como: Aris M. Ouksel, Channah F. Naiman, Clement Yu, Wei Sun, Son Dao, Ramanathan V.Guha, entre otros.

### 2.2.1 Nuestra definición de contexto

Nuestra representación del contexto tiene como único objetivo determinar la acepción de cada clase. Conociendo el significado de la clase tendremos información para decidir si existen otras clases similares. Desde el punto de vista funcional, en una ontología no existen las circunstancias de la comunicación -no hay comunicación- por lo que no hace falta representar las propiedades funcionales. Simplemente, disponemos de un modelo de representación con unas interpretaciones elegidas por el diseñador. Desde el punto de vista cognitivo, al existir una representación también existe información contextual y relaciones entre éstos contextos: subsunción y equivalencia.

Para este trabajo no hemos necesitado modelar una estructura especial llamada contexto. Hemos usado la propia representación como contexto, no suponemos más información de la que ya hay representada y de la que podamos extraer de los recursos externos. Es decir, son los propios axiomas los que definen el contexto y son las acepciones de las clases las que proporcionan la interpretación. Los recursos externos proporcionan el posible repertorio de acepciones. Cada clase tiene su propio contexto definido por su interpretación y situado en el dominio por el resto de elementos relacionados directa o indirectamente. El dominio de la representación lo define el diseñador y lo hemos considerado como un contexto general. Como el contexto de cada clase está definido por su acepción, y la acepción depende del contexto entonces son las relaciones con las demás clases y sus correspondientes acepciones las que definen su acepción. Obviamente, esto ocurre para todas las clases ambiguas, las que presentan más de una posible acepción. Las clases con una acepción tienen definido el contexto por su definición y permiten con mayor grado de propagación asentar la acepción de aquellas clases directamente relacionadas.

De esta primera observación vamos a utilizar la noción de clases relacionadas para obtener el *contexto* de la clase. Las clases están relacionadas con palabras y sus funciones, es decir, con otros elementos de la ontología mediante relaciones semánticas. Aquellas clases altamente relacionadas juegan un papel más importante que los elementos aislados. Son elementos conectores de información, predominan las relaciones e influyen en la interpretación del dominio. Los elementos aislados disponen de un contexto y significado difícil de descubrir. Como por ejemplo, decir *fuego* sin ningún otro tipo de información relacionada hace ambigua su interpretación. Volviendo a las teorías cognitivas, aunque la palabra esté aislada en la representación, ésta está en la representación y presenta algún tipo de vinculación con el resto, por lo que podemos decidir una posible acepción dentro del abanico de posibilidades. Sin embargo, la asertividad de la acepción presenta más motivos para cuestionar su certeza.

Para determinar la acepción de una clase vamos a utilizar aquellos elementos relacionados directamente o indirectamente con ella. Es este grupo de elementos los que definirán el contexto de esa clase mediante correspondencias con recursos externos. En nuestro caso utilizaremos WordNet [44; 108]. Los tesauros, como WordNet, ofrecen a parte del significado, un conjunto de relaciones semánticas con otras acepciones: sinónimos, antónimos, hiperónimos, etc. que permiten establecer relaciones directas con los constructores de la ontología. Este tipo de alineamientos ofrecen criterios comparativos entre la estructura de una ontología y la estructura de WordNet. Lo cual permite tener mayor información en la selección de una acepción. En el siguiente capítulo, veremos el tipo de técnicas existentes para seleccionar la acepción de una palabra.

## 2.3 Descubrimiento de información y otras disciplinas relacionadas

A partir de ahora, introducimos un enfoque más práctico del desarrollo de técnicas de computación de las teorías lingüísticas citadas anteriormente y las iremos relacionando con disciplinas que han ido poco a poco gestionándolas o englobándolas.

Sustituir dos términos de diferentes representaciones es a nivel computacional un problema de interoperabilidad. La necesidad de solucionar el problema de interoperabilidad apareció con las primeras representaciones de datos, como son los simples registros aritméticos, y se ha ido reproduciendo a medida que la complejidad de las representaciones iba creciendo. Una de las disciplinas donde la interoperabilidad, entre otras tareas, es esencial para el resto de procesos recibe el nombre de *descubrimiento de información* (*Knowledge Discovery*, KD).

Descubrir información es identificar patrones relevantes de datos y con posible potencial para ser utilizadas en múltiples áreas donde complementan o asisten a la toma de decisiones: en negocios, en campañas de publicidad, en descubrimientos en biomedicina, química y otras disciplinas, en mejorar las búsquedas web, en predicciones económicas, etc. El gran número de fuentes de información hace que la interoperabilidad sea un proceso clave para aumentar la eficacia de la aplicación [121].

El proceso de KD sigue una secuencia de pasos en común ampliamente utilizada. Todas las propuestas comienzan por el proceso de (i) *definición de la fuentes de datos* su elección y acceso. Un proceso de manipulación y selección de información requiere de un proceso de (ii) *filtrado*, donde en función de una serie de criterios se descarta y se prepara la información considerada útil por alguna tarea supervisada o no. Estos datos computables son suministrados a un grupo de procesos para posibles clasificaciones, agrupamientos, búsqueda de patrones, de reglas, etc. conocido como (iii) *minería de datos*

(*data mining*). Para facilitar la interpretación de los nuevos datos es conveniente (iv) *visualizar*, y para garantizar la calidad y la confianza de los mismos es necesario (v) *evaluar*.

El proceso de minería de datos es considerado como una disciplina en sí, por ser una de las etapas más complejas y donde más técnicas se aglutinan. Por este motivo, suele ser habitual confundir el proceso de minería de datos con el proceso de descubrimiento. Las técnicas aplicadas dependen del tipo de fuente: imagen, audio o texto. Esta última es la más representativa por ser el canal más utilizado en los sistemas informáticos aunque con los últimos adelantos tecnológicos en transmisión de imagen y sonido el número de estas técnicas ha crecido considerablemente. Cuando las fuentes de información son textuales la minería de datos se conoce con el nombre de minería de texto (*text mining*) [43].

La minería de texto se caracteriza por las operaciones de preprocesado textual, centradas en la identificación y en la extracción de características léxicas en documentos representados mediante algún tipo de estructura, generalmente, basados en el lenguaje natural. La mayoría de las técnicas se centran en el lenguaje natural por ser la estructura universalmente utilizada en la mayoría de los documentos físicos: documentos oficiales, históricos, leyes, libros, apuntes, contratos, etc. Estas operaciones permiten convertir colecciones de documentos en formatos explícitos estructurados posibilitando su procesado computacional y dando lugar al establecimiento de medidas con las que realizar procesos de descubrimiento. Este tipo de técnicas son aglutinadas en una área llamada procesamiento del lenguaje natural (*natural language processing*, NLP) [74; 99].

Los procesos del lenguaje natural son también utilizados en técnicas de recuperación de información (*Information Retrieval*, IR) [98; 162]. En este tipo de técnicas se requiere conocer e interpretar la estructura y el contenido para poder responder a peticiones. La riqueza expresiva, la flexibilidad de los constructores, la construcción del mensaje bajo la interpretación del emisor, entre otros factores dan a entender la dificultad de analizar un texto.

En cualquier búsqueda de información se recurre a técnicas de IR para mejorar los tiempos de respuesta y la precisión de los resultados. Como ejemplo citar un problema del libro [98] (pág.3) donde se quiere conocer *aquellos libros de Shakespeare que contienen los personajes de* Brutus ***y*** Cesar ***pero no*** Calpurnia. La solución propuesta se basa en recorrer todos los libros buscando las palabras claves en cuestión bajo la lógica de la petición. Tal como comentan los autores, hay una serie de fenómenos que serían necesarios mejorar como son las técnicas de acceso optimizadas para manejar grandes volúmenes de información, las operaciones flexibles de mapeo para proveer soluciones próximas a óptimos aceptables y agrupaciones de resultados para mejorar la respuesta entre todas las posibles soluciones.

En este tipo de problemas, hay que remitirse a reglas gramaticales y sintácticas para poder tener unas pautas de procesamiento. Por ejemplo, al analizar una palabra se ha de tener en cuenta la colocación dentro del texto

ya que influye en teorías contextuales y gramaticales, en medidas estadísticas (frecuencia, dispersión, colocación espacial, etc.) y, habitualmente, es necesario saber su significado para poder responder eficazmente a peticiones o a predicciones. Estas últimas reciben el nombre de desambiguación del sentido de la palabra (*Word Sense Disambiguation*, WSD) [1; 43; 147].

En la mayoría de las tareas de cualquier proceso de descubrimiento (construcción de índices, ponderación de términos, evaluación, clasificación Naive Bayes, etc.) existen procesos de *comparación* de términos más o menos exigentes en cuanto a representación e integración acorde a la naturaleza del problema. La comparación de términos o el mapeado de términos está presente en cualquiera de todas estas disciplinas que hemos ido nombrando superficialmente. En definitiva, el conjunto de técnicas aplicadas al mapeado de ontologías son técnicas de descubrimiento de información en representaciones semánticas para lo cual se requieren técnicas de procesamiento del lenguaje natural para identificar los elementos, interpretarlos y manejarlos adecuadamente acorde a la naturaleza del problema.

### 2.3.1 Características de los modelos de representación

Tal como hemos comentado, mapear ontologías difiere a otras técnicas de mapeado por el lenguaje utilizado. El lenguaje juega un importante papel por la capacidad de suministrar medidas con las que podamos manejarlo y aplicar técnicas ajenas a él. Para entender con más precisión el papel que desempeña el lenguaje de la representación, vamos a caracterizarlo según una serie de características que afectan directamente al mapeado. Además, para ilustrar cada una de ellas usaremos una serie de ejemplos basados en tres tipos de modelos de representación, ampliamente utilizados, como son: los modelos de entidad relación (ER), modelos orientados a objetos (OO) y modelos ontológicos (OWL[3]).

- Expresividad. La expresividad es la capacidad de representar ideas lo más realistas posibles con una gran viveza de detalles. Son los propios constructores del lenguaje que restringen la expresividad de las ideas. La expresividad del lenguaje no está relacionada con la bondad del mismo. Los lenguajes naturales disponen de constructores capaces de definir nuevas ideas que no existen en el propio lenguaje. Los lenguajes artificiales suelen adolecer problemas de expresividad por la limitación de los constructores para definir dominios computables. Cuando se requiere incrementar la información, se añaden nuevos constructores generalmente ajenos a la representación original. De esta manera, en modelos ER los constructores son tablas e identificadores de atributos, no existen las relaciones entre tablas. Podemos considerar los disparadores (*triggers*) como añadidos para incrementar la definición del modelo. En modelos OO podemos definir

[3] Usamos el acrónimo de OWL por ser el lenguaje referencia de las representaciones en la web semántica

relaciones de herencia entre objetos. En modelos OWL podemos definir relaciones de transitividad entre clases. Sin embargo, una característica no representable en OWL bajo su modelo lógico es la incertidumbre.

- Ambigüedad. La ambigüedad es una idea abstracta de la posibilidad, de la duda, incertidumbre o confusión en entender de varios modos o admitir diferentes interpretaciones de una idea. Una comunicación totalmente ambigua conlleva unos resultados inesperados y pocos fructuosos. Por tanto, el objetivo de cualquier lenguaje es ser lo menos ambiguo posible. El lenguaje natural se caracteriza por el gran número de términos que presentan múltiples acepciones, pero el propio contexto de la comunicación suele dejar por sentado cual es la acepción. El contexto de los elementos de una representación determina el nivel de ambigüedad de una representación. En los tres modelos está presenta la ambigüedad en diferente grado pero por lo general depende en gran parte por el diseñador del modelo. Es decir, el mensaje puede ser ambiguo independientemente del lenguaje utilizado. Por ejemplo, la tabla *planta*, el objeto *planta* y la clase *planta* sin ningún tipo de vinculación con otro elemento presenta el mismo nivel de ambigüedad en las tres representaciones. Los constructores del lenguaje delimitan la ambigüedad en función de su expresividad, cuanto más expresivo sea un lenguaje -y se haya hecho uso de esta riqueza- más detalles o medidas dispondremos para decantarnos por una acepción.
- Estructuración. La estructura básica de una idea en un lenguaje natural es la frase, donde se distinguen tres partes principales: sujeto, verbo y complementos. Las frases se agrupan en párrafos, citaciones, secciones, etc. se alteran en preguntas o exclamaciones y el sujeto o los complementos pueden contener frases subordinadas. Todas estas posibles combinaciones complican la detección de cada uno de los elementos. Por el contrario, para simplificar el procesamiento de datos en los anteriores modelos la estructura suele ser bastante rígida y delimitada. La estructura posibilita el uso de ideas ya existentes, el acceso a las mismas, etc. La estructura depende de la sintaxis elegida para representar el modelo. Con una sintaxis XML podemos representar modelos ER, OO y OWL con más o menos eficiencia.
- Acceso. El acceso define las mínimas unidades identificables, accesibles y observables del modelo. La estructura, y la sintaxis, influye en la porción del elemento accedido. Por ejemplo, en un modelo ER accedemos a tablas y a valores; en un modelo OO accedemos a los objetos e instancias de estos; y en un modelo OWL accedemos a clases, restricciones, propiedades e individuales.
- Economía lingüística. Se entiende como la capacidad de expresar información con la mínima representación posible. Es un factor que depende de la expresividad y de los constructores del lenguaje. Por ejemplo, si no existiera la palabra 'rápel' para describirla utilizaríamos otro conjunto de palabras "descenso rápido en el que se utiliza una cuerda por la que se desliza el alpinista". Ambas ideas representan lo mismo pero una inclu-

ye más términos que otra, una representación es más económica, menos ambigua y fácilmente computable.
- Modelo de interpretación. Es ajeno a este trabajo definir el modelo de interpretación del lenguaje natural [17; 18]. Simplifiquemos el modelo de interpretación como la capacidad de comprender las implicaciones de una idea bajo un contexto. Es por tanto, conocer el significado y sus consecuencias. Si gritáramos *fuego* en una sala de cine todos los asistentes interpretarán el mensaje bajo ese contexto actuando en consecuencia. El modelo de interpretación no está definido en los modelos ER. En los modelos OO existe la noción de herencia y en los modelos OWL es explícito. Si los constructores de un lenguaje se basan en un modelo de interpretación se disminuye la ambigüedad de los elementos.

Los modelos semánticos basados en ontologías frente a modelos tradicionales basados en ER o OO son más expresivos; poseen un modelo de interpretación o modelo lógico que disminuye la ambigüedad de la representación; la estructuración facilita tareas computacionales y un mayor acceso a los elementos independientemente de su definición; facilitan la reutilización de definiciones y, finalmente, posibilitan, gracias al modelo lógico, una mayor economía lingüística. Por el contrario, el modelo lógico subyacente incrementa la complejidad de la modelización, del desarrollo y de la explotación frente a estos dos modelos [15].

### 2.3.2 El proceso de mapeado

El mapeado de ontologías (*Ontology Mapping*, OM) es un proceso por el cual se descubren correspondencias entre dos términos de diferentes representaciones. El tipo de correspondencia o relación entre dos términos es llamado alineamiento. Cada alineamiento consta de un identificador por cada elemento implicado y de un tipo de información sobre el tipo de relación. Tipos básicos de relación son la igualdad, la diferencia y la subsumisión. Por ejemplo, igualdad, entre `car` y `vehicle`; diferencia, entre `hot` y `cold`; y subsunción, entre `car` y `vehicle`. De la misma manera que la representación está sometida a la interpretación, los alineamientos están sometidos bajo la interpretación de una persona, en el caso de un sistema supervisado o por unos criterios dinámicos, en un sistema sin supervisar.

Muchas de las aproximaciones siguen un proceso común [22; 51; 140]. Algunas aproximaciones cambian o combinan diferentes pasos pero los principios son los mismos. Nuestra particular síntesis de cada uno de ellos, inspirada en el proceso de descubrimiento, está representada en la figura 2.1 y contiene las fases siguientes:

1. **Entrada (*Input*)**. Dos o más ontologías son las fuentes de datos del proceso. Si un algoritmo ofrece resultados en el mismo formato que la entrada, entonces puede mapear múltiples ontologías en combinaciones

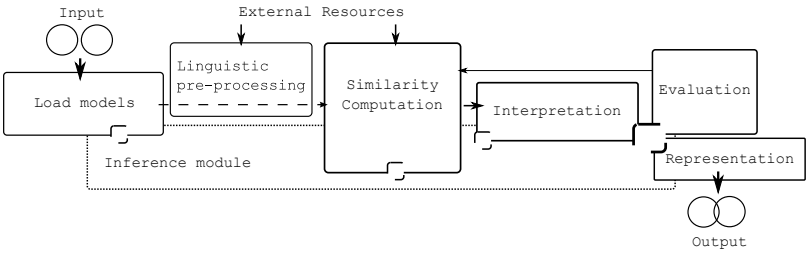


Figura 2.1: Fases del proceso del mapeado de ontologías

de dos a una. En esta fase se asegura el acceso a las ontologías y a sus importaciones si acontece.

2. **Carga de modelos (*Load Models*)**. Las ontologías son una combinación de múltiples elementos, de reglas y de otras ontologías que el sistema necesita manipular. Este proceso depende del modulo de inferencia para manipular los elementos implícitos.
3. **Preprocesado lingüístico (*Linguistic Pre-Processing*)** El nombramiento de elementos al ser realizado por personas es complejo de gestionar y requiere de un preprocesado lingüístico. En él se suelen llevar a cabo los siguientes procesos: *tokenization*, eliminación de términos comunes, normalización, analizador lingüístico (*stemming*), tratamiento de diacríticos, *capitalization* y otros usos específicos de la lengua de representación (fechas, horas, notación económica, etc.). Son de vital importancia ya que influyen directamente en el resto de procesos. A continuación se explican algunos de los procesos más habituales en la minería de texto. En los siguientes capítulos veremos cada proceso adaptado al mapeado de ontologías.
   - *Tokenization*. Es la tarea de identificar piezas o *tokens* en el nombre. Esto se debe al uso de nombres compuestos o composiciones de palabras para dotar de mayor expresividad al elemento nombrado. Suele ser habitual encontrarse con dos tipos de separadores: el uso de símbolos como guiones, punto u otros caracteres y el uso de la nomenclatura *CamelCase*. *CamelCase*[4] es una convención de nomenclatura ampliamente extendida en múltiples lenguajes de programación. En el caso de las ontologías, uno de los manuales[5] más conocidos, creado por la universidad de Manchester para el aprendizaje del editor Protégé, aconseja tal nomenclatura para el nombramiento de clases y propiedades. En la práctica, las palabras son unidas sin espacios en blanco, donde cada letra inicial está capitalizada dentro de la composición. La primera letra de la composición es indiferente si lo está o no. En el caso de las ontologías, la primera letra de la clases tiende a capitalizarse

[4] http://c2.com/cgi/wiki?CamelCase
[5] http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/

y en una propiedad, no. Algunos ejemplos son: `iPod, McDonald's, pizzaToping, WhiteNonSweetWine, hasTopping`, etc.
- Eliminación de términos comunes. Algunas palabras suelen poseer poco valor para posteriores tareas y deben de ser excluidas. Estas palabras reciben el nombre de *stop words* y suelen ser preposiciones, artículos y en ocasiones, adverbios y conjunciones. La estrategia más usual es su eliminación mediante su identificación en una lista de palabras *stop words*. Algunos ejemplos son: `the`, `a`, `for`, `by`, `since`, `whilst`, etc.
- Normalización. Es el proceso de identificar canonizaciones parecidas en la secuencia de caracteres de la palabra. Como por ejemplo, para detectar la similitud entre `AIDS` y `A.I.D.S`.
- *Stemming* y *lemmatization*. El objetivo de ambos procesos es encontrar la base común de una palabra, el lema, a causa de los cambios gramaticales (número, persona, tiempo u otro morfema derivativo). Algunos ejemplos son: {`am`, `are`, `were`} = `be` y {`organizing`} = `organize`.
- Diacríticos. El inglés no se caracteriza por este aspecto, pero si posee el genitivo sajón. En documentos escritos es habitual encontrarse con este caso pero no entraremos en su gestión pues es ajeno al nombramiento de elementos ontológicos.
- Capitalización (*Capitalization*). Es el proceso de convertir todos los caracteres de una palabra a minúsculas -generalmente- o a mayúsculas. Cuando la palabra es un nombre propio o un acrónimo se puede alterar el significado de la misma, por ejemplo, el caso de (George) `Bush` y el caso del acrónimo `C.A.T`.

4. **Recursos externos (*external resources*)**. Es habitual obtener más información a partir de recursos lingüísticos como diccionarios o tesauros, de otras ontologías o de resultados previos para complementar el proceso.
5. **Computación de la similitud (*similarity computation*)**. Es el proceso central donde se combina toda la información con el objetivo de encontrar alineamientos. Este punto se desarrolla en el siguiente capítulo mediante los trabajos actuales en esta área.
6. **Interpretación (*Interpretation*)**. Cuando se descubre una relación entre dos elementos, independientemente de su representación, el algoritmo ha de determinar que tipo de relación guardan los dos elementos -su interpretación- y ha de integrarla en los modelos originales. En un primer ejemplo supongamos que existe una vinculación entre la palabra `car` y `plane` mediante la palabra `vehicle`. Esta vinculación se ha de interpretar, en este caso como superclase de ambas pero a la vez se ha de integrar, es decir, se ha de poder responder a preguntas como ¿existe la palabra `vehicle` en algún modelo?, ¿`car` y `plane` pueden compartir la misma superclase? y ¿puede producir alguna incoherencia? Supongamos otro caso entre las palabras `car` y `truck`. En ellas existe un número mayor de características comunes que entre `car` y `plane`. De alguna manera se ha de

poder representar esa mayor similitud. Para tales casos, la mayoría de las propuestas actuales utilizan un valor numérico entre 0 y 1.

7. **Evaluación (*evaluation*)**. Mediciones internas y externas de calidad, junto con resultados de rendimiento se deben de proveer para posibilitar un refinamiento automático y una posterior comparación con otros algoritmos.
8. **Representación (*representation*)**. La representación de los alineamientos ha de realizarse con el objetivo de poder ser utilizados después de la finalización del algoritmo e integración de los mismos acorde a la naturaleza de la aplicación. Cuando el algoritmo es supervisado es recomendable representar gráficamente los mismos.
9. **Modulo de inferencia (*Inference Module*)**. El modelo lógico en las ontologías hace que sea indispensable trabajar con un razonador. Es una fase omnipresente para el resto. El razonador infiere hechos implícitos que de ninguna otra manera podríamos obtenerlos. Generalmente, estos nuevos hechos son basados en la pertenencia a clases y a relaciones de jerarquía.
10. **Salida (*Output*)**. La salida ha de representarse con algún lenguaje para facilitar tareas computacionales. Es una fase ligada a la representación pero depende de la naturaleza de la aplicación que requiere del algoritmo de mapeado. Por lo general, para facilitar una correcta compatibilidad con las herramientas se debe de usar el mismo lenguaje que el de las fuentes.

## 2.4 Sumario

Desde la perspectiva filosófica, léxica y computacional establecer una similitud es encontrar aquellos elementos que intercambiados en ambos modelos de comunicación mantienen la veracidad del mensaje. Desde la perspectiva filosófica, hay dos maneras de lograrlo: identificar mediante una definición los elementos o realizar un cálculo de similitud en función de las características lógicas. En la representación, los elementos poseen unas características lógicas que las decide el creador del mensaje por criterios culturales, personales o funcionales, dando por hecho un conocimiento implícito: el contexto. El contexto define cada acepción de los elementos, y el entorno lingüístico de la comunicación. El contexto y el lenguaje determinan las características lógicas de los elementos y la definición de los mismos. Al mismo tiempo, la expresividad y el tipo de constructores del lenguaje influyen en la definición implícita del contexto haciéndolo más o menos manejable a nivel computacional.

Las técnicas de mapeado independientemente del nivel del lenguaje que manejen se pueden clasificar como técnicas propias de la minería de datos. Dentro de estas técnicas existen categorías específicas como la minería de texto, el procesamiento del lenguaje natural y la desambiguación de palabras que han tratado el contexto como una fase importante para la resolución de algunos de sus objetivos. Las técnicas de minería de datos son usadas

para la extracción de información y englobadas en una inmensa área como es la inteligencia artificial. La similitud de elementos semánticos mediante ontologías requiere de los mismos principios básicos que cualquier técnica de mapeado.

En la figura 2.2 vemos una representación de todos aquellos factores y disciplinas relacionadas con las técnicas de similitud con el objetivo de comparar dos palabras. Cada palabra está influenciada por su contexto y a su vez, éste está definido por los diferentes perfiles del lenguaje, del emisor y del receptor. Las hipótesis que han influido en el campo de la computación han sido las teorías filosóficas y lingüísticas.

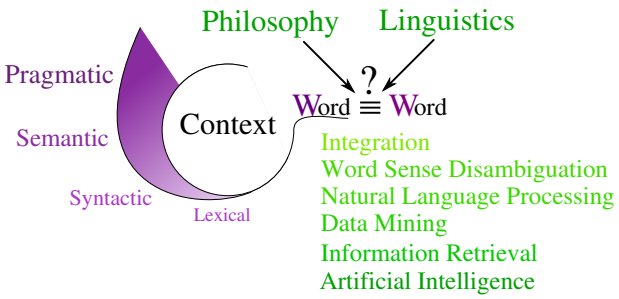


Figura 2.2: Disciplinas y conceptos relacionados

De manera simbólica citamos otras áreas donde la aplicabilidad de las técnicas de similitud desempeñan un eje central en sus correspondientes prestaciones [160]: *language modelling, automatic thesaurus generation, PP attachment ambiguity resolution, conjunction scope identification, anaphor resolution, collocation extraction, word sense separation, text simplification, metonymy resolution, compound noun interpretation, learning countability preferences, topic identification* y *spelling correction.*

En el siguiente capítulo se detallan las técnicas y los trabajos relacionados. La mayoría de las propuestas no han considerado la semántica del modelo, donde destacamos la ausencia del significado como medida básica de comparación.

# 3

# Trabajo relacionado

En este capítulo presentamos las técnicas más representativas de las fases del mapeado para continuar con las propuestas más relevantes de la área. Para concluir, realizamos un análisis comparativo de todas las propuestas y un debate respecto al conjunto de medidas requeridas para la realización de un mapeado semántico

## 3.1 Tareas del mapeado

Desde nuestro punto de vista, las tareas o funciones de las diferentes propuestas [41; 48; 67; 84; 112; 126; 159] las hemos sintetizado en una serie de etapas, las cuales son: carga, preprocesamiento, cálculo de similitud, interpretación de los resultados, evaluación y representación (ver figura 3.1). La explicación gira alrededor de cada una de estas fases. En comparación con nuestra propuesta, ésta sí incluye el proceso de integración mediante la representación de alineamientos y, también incluye el proceso de inferencia, como proceso clave pare el descubrimiento de nuevos hechos y relaciones. Las fases de nuestra propuestas se sintetizan en la figura 2.1.

El primer paso del proceso es el tratamiento léxico de los términos. Ambas ontologías presentan estilos y nomenclaturas que difieren en el nombramiento de elementos lo cual dificulta la comparación de los mismos. Posteriormente, el cálculo de similitud se realiza por un conjunto de medidas en algunos casos independientes y en otros, no; que, además, son computadas paralelamente y/o secuencialmente. Esta flexibilidad depende de la heurística y de las medidas utilizadas. Las medidas se clasifican en dos tipos sintácticas y estructurales siguiendo las clasificaciones de [41] y de [126] (ver figura 3.2). Algunas de estas medidas requieren de información externa para contrastar y calcular el grado de similitud. La mayoría de estos recursos externos consisten en diccionarios, tesauros, webs con información estructura como la Wikipedia (DBpedia), *Unified Medical Language Systems*, ontologías genéricas (*upper-ontologies*) y/o resultados previos para mejorar los resultados, ya que

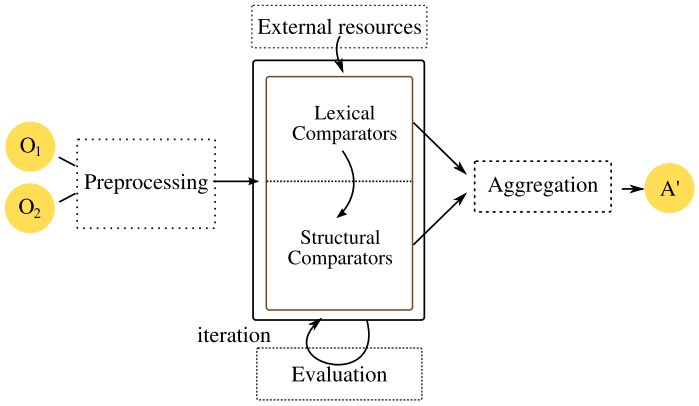


Figura 3.1: Simplificación de las tareas en el proceso de OM

es posible tener varios al mismo tiempo. Algunas propuestas integran módulos de refinamiento iterativo donde en cada paso o medida se mejora la precisión de los alineamientos. La fase de agregación, la que gestiona la estrategia del algoritmo, permite combinar todas las medidas en una única decisión. La evaluación tiende a ser dinámica pero como norma se contrastan los resultados con una muestra creada por expertos. Según la propuesta, la fase de evaluación puede estar por delante o no de la fase de agregación aunque pueden existir procesos de evaluación independientes para cada una de las medidas. Finalmente, el alineamiento ha de presentar un formato para facilitar la posterior utilización o integración de alineamientos.

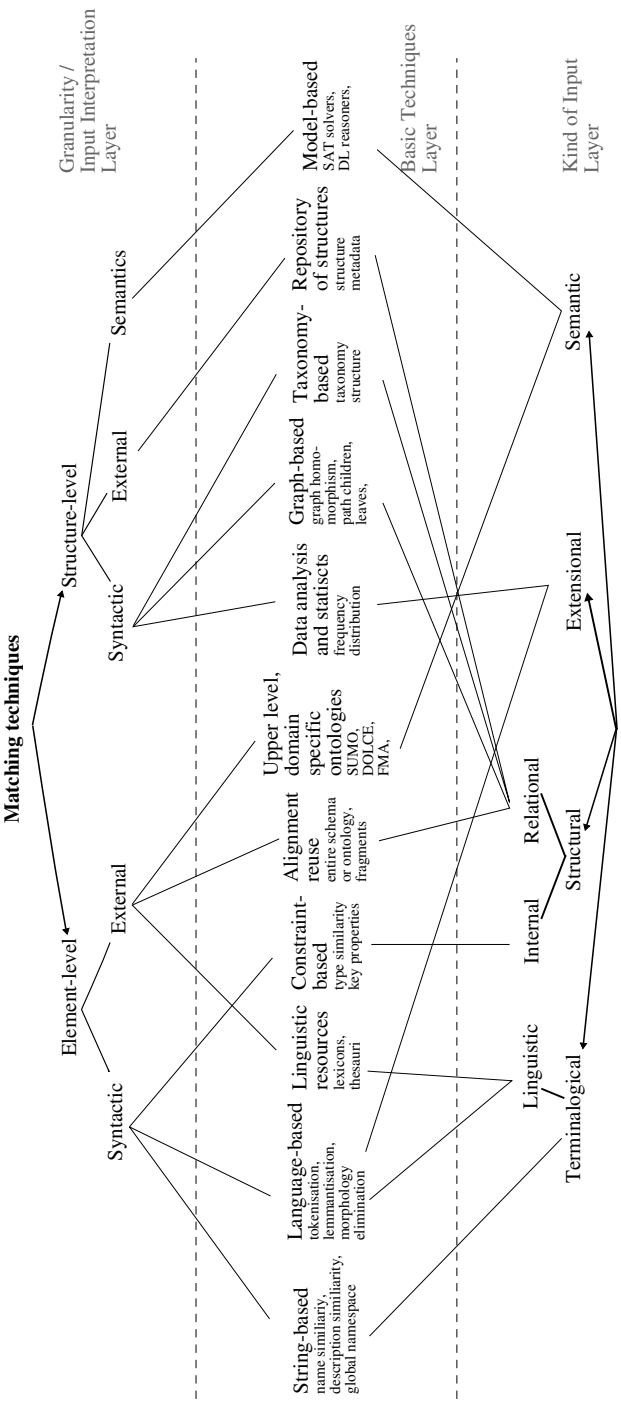


Figura 3.2: Clasificación de las técnicas de mapeado

A continuación, introducimos los tipos de medidas, de estrategias, de recursos, de evaluaciones y de representaciones que forman parte de esta disciplina y de cada uno de los trabajos relacionados.

## 3.2 Preprocesado

En la anterior sección 3 se explica el concepto de preprocesado lingüístico de palabras. Su objetivo es facilitar la manipulación computacional de textos. La mayoría de estas técnicas provienen de propuestas ampliamente aceptadas en el procesamiento del lenguaje natural. Diferenciamos los analizadores lingüísticos básicos y aquellos entornos que aglutinan e implementan estos analizadores y funciones básicas del tratamiento de palabras (*tokenizadores*, indexación, etc.). Vemos a continuación los diferentes analizadores y entornos para desempeñar está función.

### 3.2.1 Analizadores lingüísticos

El objetivo de una analizador lingüístico (*stemmer*) es encontrar el lema, la parte invariante de una palabra, eliminando sufijos. Los principales analizadores son:

- **Analizador de Lovins**. Fue el primer algoritmo presentado en 1968 por Julie Beth Lovins consiguiendo una gran referencia en posteriores trabajos. Consiste en una serie de reglas diseñadas para cubrir las excepciones comunes categorizadas en cada una de las posibles 297 terminaciones. Todas las terminaciones son asociadas con una excepción común: la reducción al menos ha de conservar dos caracteres. Se eliminan caracteres en función de un principio de longitud. Durante su desarrollo se descubrió que pocos ejemplos de tales reglas podrían ser aplicados. En cada terminación existe un número especial de casos que causa errores en la obtención del término base.
- **Analizador de Porter**. Fue presentado en 1980 y desarrollado por Martin Porter en la Universidad de Cambridge. Es ampliamente usado y dispone de múltiples implementaciones en diferentes lenguajes. El algoritmo consta de 6 pasos. El primero gestiona los pasados regulares y los plurales. El segundo elimina prefijos de gerundio y ciertos sufijos. El tercer paso transforma las terminaciones en 'y' en 'i'. El resto de pasos contienen reglas específicas para sufijos comunes. El algoritmo original fue publicado en 1979 en el libro de *Information Retrieval* [154]. Un año más tarde fue publicado en [124] y posteriormente en [72].
- **Analizador Paice/Husk**. El analizador Paice/Husk, también conocido como algoritmo de Lancaster, fue desarrollado por Chris Paice en la Universidad de Lancaster [1] en los 80, y fue originalmente implementado con

[1] http://www.comp.lancs.ac.uk/computing/research/stemming/index.htm

la ayuda de Gareth Husk. Es un algoritmo iterativo que va eliminando las terminaciones en un número indefinido de pasos. La eliminación se basa en reglas agrupadas por los caracteres de terminación y son aplicadas en secuencia si atañe. El analizador ha sido implementando en Pascal, C, PERL y Java.

- **Analizador UEA-Lite**[2]. Similar al resto de analizadores, opera con un conjunto de reglas usadas en dos diferentes fases: en la primera se eliminan *tokens* y en la segunda se eliminan sufijos [65].

Respecto al rendimiento de estos analizadores, [120] realizó un estudio respecto a los tres primeros. Las conclusiones fueron las siguientes. El algoritmo de Porter es un analizador ligero, es decir, el número de palabras a reducir no es elevado. El algoritmo de Lovins es considerado moderado, y el algoritmo Paice/Husk es considerado pesado. Cuanto más pesado es un analizador más innecesariamente interviene en la reducción de palabras. Destaca ciertas evidencias por las que el algoritmo de Lovins es más propenso a errores que los otros dos. La diferencia de intervenciones entre Porter y Paice/Husk es tan grande que es insignificante la comparación de su precisión.

Los autores de UEA-Lite, un analizador más actual, realizaron un análisis comparativo con estos tres analizadores [65]. UEA-Lite es considerado un analizador ligero parecido al analizador de Porter difiriendo ligeramente en la calidad de la reducción al conseguir un mayor número de palabras escritas correctamente. Hay que tener en cuenta que la reducción de una palabra por una analizador puede generar una palabra sin sentido lo cual puede devenir en la calidad de procesos posteriores. Por ejemplo, podrían reducir la palabra `stemming` en `ste`.

En las propuestas no se incluyen algoritmos de reducción al lema (*lemmatisation*) ya que generalmente la búsqueda del lema se da en verbos. La aparición de verbos en ontologías es ocasional exceptuando en el nombramiento de propiedades. Las propiedades describen las características y las acciones de los objetos. Un caso típico es la partícula `has`. Es fácil encontrar está partícula formando parte del nombre de las propiedades, algunos ejemplos: “hasAuthor”, “hasPassport”, “hasName”,....

### 3.2.2 Entornos de trabajo

Existe un determinado grupo de entornos de trabajo que unifican tareas de NLP y de esta manera, facilitan tareas de análisis léxico, filtrado de palabras, gestión de frases, etc. Listamos aquellos de interés:

- **GATE**[3] es un código abierto que aglutina rutinas capaces de solventar problemas en la manipulación de texto. Las principales funciones son el

[2] http://lemur.cmp.uea.ac.uk/Research/stemmer/
[3] http://gate.ac.uk/

modelado y almacenamiento de estructuras especializadas en datos, medición, evaluación, *benchmarking*, visualización, edición, manipulación de ontologías, de árboles u otras estructuras, extracción de estancias para procesos de aprendizaje y facilita la incorporación de otros sistemas de aprendizaje como WEKA, YALE, SVM Lite,...

- **MontyLingua**[4] es un aplicativo gratuito, construido para manipular texto. Utilizado en técnicas de *Information Retrieval*: extracción y *queries*. Entre las numerosas funciones es capaz de extraer de una frase: el sujeto, verbo, predicados, adjetivos, complementos, frases subordinadas, eventos, fechas, horas y otro tipo de información.
- **MorphAdorner**[5] es un programa basado en línea de comandos en Java que actúa de gestor de tareas. Estas son capaces de realizar cambios morfológicos en las palabras o filtrar información de adorno en textos.
- **OpenNLP**[6] es un conjunto de librerías que aglutinan diferentes proyectos de código abierto relacionados con NLP. Contiene una variedad de herramientas basadas en Java las cuales realizan tareas de *tokenization*, detección de partes de una sentencia, etiquetado, segmentación, conversión y detección entre otras tareas.
- **Natural Language Toolkit** (NLTK)[7] es un conjunto de módulos en código abierto en Python para la investigación y el desarrollo de técnicas relacionadas con NLP.

## 3.3 Medidas léxicas

Una vez finalizadas las tareas de preprocesado, el siguiente paso es determinar la similitud de los elementos. Para establecer una similitud necesitamos considerar aquella información que permita discernir cualidades específicas. Está información es considerada una medida. La mínima unidad de representación son las palabras que definen los conceptos, unidades léxicas. Por tanto, a nivel léxico, podemos considerar dos tipos de medidas, las que analizan la similitud entre caracteres y las que consideran un conjunto de palabras vecinas. Estas últimas suelen requerir de recursos externos para encontrar esta relación de vecindad, como pueden ser grupos de sinónimos o de antónimos.

### 3.3.1 Distancias simples entre palabras

Las funciones de distancia comparan un par de palabras para obtener un valor numérico del grado de similitud, equivalencia o sinonimia. A este tipo de medidas se les denomina distancias. La mayoría son distancias de edición siendo

[4] http://web.media.mit.edu/ hugo/montylingua/
[5] http://morphadorner.northwestern.edu/
[6] http://incubator.apache.org/opennlp/
[7] http://www.nltk.org/

útiles en sugerir correcciones ortográficas en editores de texto. Estas distancias de edición contabilizan el número y el coste de operaciones de edición, inserción, eliminación y sustitución de un carácter. Estas operaciones mutan caracteres de una palabra para obtener otra.

- Distancia de Hamming [55] permite detectar la diferencia entre dos palabras mediante el cálculo del número de cambios entre caracteres para obtener la otra palabra. Por ejemplo, la diferencia entre `casa` y `cama` es de 1. Sólo se aplica a palabras o mensajes con la misma longitud.
- Distancia de Levenshtein [92] también conocida como distancia de edición -por ser la más utilizada-, funciona de la misma manera que la distancia de Hammming pero admite palabras de diferente tamaño mediante las operaciones de inserción o eliminación.
- Distancia de Needleman-Wunch [115] usado comúnmente en bioinfomática para mapear proteínas o secuencias de nucleótidos, realiza un alineamiento global entre secuencias con un coste de penalización manual. Por ejemplo, en la siguiente matriz están las diferencias entre los elementos y se necesita calcular la similitud de dos fragmentos: GGTC y GT-C, con una penalización por carácter de -3, el resultado se puede ver en la fórmula 3.1.

| | G | T | C |
|---|---|---|---|
| G | 1 | -3 | 4 |
| T | 0 | 1 | -2 |
| C | -2 | 1 | 4 |

$$s(GGTC, GT{-}C) = s(G,G){+}s(G,T){+}(1,3){+}s(C,C) = 1{-}3{-}3{+}4 = -1 \tag{3.1}$$

  La diferencia de términos ya ha sido determinada por una aplicación externa a un recurso lingüístico pero esta distancia se puede aplicar con palabras compuestas, secuencias de palabras o partes de la misma palabra.
- Distancia de **Monge-Elkan** [111] está basada en la similitud de los diferentes trozos o *tokens*. Determina la similitud de cada uno de los *tokens* y obtiene el valor medio de la suma ponderada por el número de *tokens*.

$$MongeElkanSim(s_1, s_2) = \frac{1}{\mid tokenize(s_1) \mid} \sum_{i=1}^{|tokenize(s_2)|} TokenSim(t_i, t_j) \tag{3.2}$$

- Distancia de Jaro [63]. Dada dos palabras $s = a_1...a_k$ y $t = b_1...b_L$ se define un carácter común si $b_j = a_i$ tal que $i - H \leq j \leq i + H$ donde $H = \frac{min(|s|,|t|)}{2}$ . Siendo $s' = a'_1...a'_k$ el conjunto de caracteres en común en t (en el mismo orden que aparecen en $s$) y siendo análogo $t' = b'_1...b'_L$, se define una *transposición* por $s', t'$ en una posición $i$ tal que $a'_i \neq b'_i$. Siendo $T_{s',t'}$ la mitad del número de transposiciones para $s'$ y $t'$. La similitud de Jaro para $s$ y $t$ es:

$$Jaro(s,t) = \frac{1}{3}\left(\frac{\mid s' \mid}{\mid s \mid} + \frac{\mid t' \mid}{\mid t \mid} + \frac{\mid s' \mid - T_{s',t'}}{\mid s' \mid}\right) \tag{3.3}$$

- Distancia de Jaro-Winkler [163] es una variante de la distancia de Jaro, donde se usa una longitud $P$ del prefijo común más largo entre $s$ y $t$. Siendo $P' = max(P, 4)$ se define:

$$Jaro - Winkler(s,t) = Jaro(s,t) + \frac{P'}{10}(1 - Jaro(s,t)) \tag{3.4}$$

- Similitud de Jaccard [60] se define como el número de elementos de la intersección dividido por el total de elementos de la unión. Esta similitud se aplica a diferentes técnicas, en este caso se aplica sobre caracteres.

$$J(A,B) = \frac{\mid A \cap B \mid}{\mid A \cup B \mid} \tag{3.5}$$

- Coeficiente de Dice [34] se define como la información compartida entre dos elementos sobre la cardinalidad de ambos.

$$s = \frac{2 \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid} \tag{3.6}$$

- Distancia SoundEx [167] consideramos este algoritmo como el representativo de un grupo de algoritmos basados en la pronunciación de las palabras. La similitud de la pronunciación de ambas palabras define la similitud de ambas palabras.
- Por simplificación del trabajo, citamos otra serie de medidas aplicadas en otras áreas con la misma finalidad: distancia de Smith-Waterman [144] y sus variantes: FastA y BLAST [104]; distancia de Gotoh [50], distancia de Hellinger [20], divergencia de Skew [87], distancia de Kendall's Tau [26], distancia de Fellegi y Sunters [45], entre otros.

### 3.3.2 Medidas léxicas usando recursos externos

Otro repertorio de medidas se basa en la comparación de conjuntos de palabras próximas. La proximidad puede basarse por sinonimia, por palabras en la definición, por frecuencia, por tramos de caracteres, en jerarquías o en algún otro tipo de relación presente en la representación o en recursos externos. Este tipo de medidas se caracterizan por crear una bolsa de palabras de manera heurística. A continuación listamos las medidas más significativas.

- Similitud por coseno. Se crea un vector con las palabras relacionadas. El problema es determinar la similitud de dos vectores o en algunos casos es encontrar la similitud de cada par de palabras en el vector. Es una medida muy popular por su sencillez. Hay dos vectores ortogonales donde un valor 0 del ángulo representa una ausencia de similitud. Cada componente del

vector se transforma mediante algún tipo de heurística en un número escalado.

$$cos(n_1, n_2) = \frac{\sum_v P(v \mid n_1)P(v \mid n_2)}{\sqrt{\sum_v P(v \mid n_1)^2 \sum_v P(v \mid n_2)^2}} \tag{3.7}$$

- TF/IDF. Intuitivamente representa un equilibrio entre las apariciones y el total de su relevancia. Por un lado, se calcula la frecuencia de un término ($t$) en el conjunto de información relacionada a una palabra o a un documento ($c$), $tf_{t,c}$. Por otro lado, se calcula la inversa por su referencia, $idf_{t,c}$. Suele utilizarse como convertidor numeral de cada tupla -par de palabras- en los vectores. Las frecuencias son por aparición en recursos externos.

$$tfidf_{t,c} = log(tf_{t,c}) \times log(idf_{t,c}) \tag{3.8}$$

- $q$-gram o $n$-gram. No es propiamente una medida de similitud que requiera recursos externos. Una palabra es dividida en una serie de secuencias de tamaño $q$, a diferencia de los *tokens* estas partículas pueden contener solapamientos. El problema se reduce en encontrar la similitud entre las partículas y en su totalidad. Al ser sencilla de aplicar resulta idónea para combinar con otro tipo de medidas para obtener numerales y, posteriormente, aplicarles técnicas de comparación como la similitud de Jaccard o TF/IDF.

Evaluaciones, extensiones y algunos ejemplos explicativos de las anteriores técnicas, tanto de las medidas de distancia simples o como aquellas que requieren de información extra, se encuentran en los trabajos siguientes: [25; 43; 113; 160; 162].

### 3.3.3 Medidas estructurales

Los constructores de las ontologías permiten estructurar la información en tripletas, taxonomías o árboles jerárquicos sin considerar ciclos en las relaciones de subsunción, o grafos considerando ciclos o las firmas de las propiedades[8]. En este trabajo diferenciaremos dos tipos de medidas estructurales, las que no involucran recursos externos y las que sí.

La mayoría de las técnicas estructurales que no emplean recursos externos se basan en técnicas de comparación de árboles y mayoritariamente de grafos. La idea en estas medidas es determinar el grado de similitud entre estructuras cercanas a ambos conceptos en sus respectivas representaciones. Esta similitud estructural define la similitud de los conceptos. En algunos casos durante este proceso se cuenta con la similitud estructural de las clases vecinas obteniendo un proceso recursivo o, en otros se tiene en cuenta un determinado tipo de relaciones como son: los individuales o las subclases. De la misma manera que

[8] La firma de una propiedad es la tripleta formada por la clase dominio, el identificador y la clase rango de la propiedad.

en las medidas léxicas la flexibilidad para combinar diferentes consideraciones hace complicado sintetizar en único listado todas las posibilidades. En la reimpresión de [96] podemos encontrar medidas básicas de comparación de grafos y en los trabajos de [73; 166] algunos casos prácticos de adaptación de estructuras XML o modelos ER a gráfos.

### 3.3.4 Recursos externos

Siempre es útil disponer de más medidas para establecer nexos en común entre los términos. Por esa razón, los recursos externos son fundamentales. En el mapeado estructural, los recursos externos son la regla de medición con la que posicionar elementos bajo una escala. Algunas medidas léxicas usan este tipo de recursos para generar la bolsa de palabras relacionadas.

Podemos diferenciar tres tipos de recursos: diccionarios, tesauros y ontologías de amplio dominio (*upper-ontologies*). En algunos trabajos se ha utilizado UMLS y Wikipedia pero la función es similar y no procedemos a su explicación. De los recursos nombrados citamos aquellos gratuitos para la comunidad científica.

#### Diccionarios

Un diccionario es un listado ordenado de términos y acepciones. Por tanto, dada un palabra podemos conocer el conjunto de significados posibles.

Para no extender la explicación de estos por su amplia disponibilidad, simplemente nombramos a la plataforma más activa llamada *DICT Development Group*[9]. Integra información de los siguientes diccionarios, tesauros y catálogos: *the Collaborative International Dictionary of English v.0.48, WordNet 2.0, Moby Thesaurus II, Elements database20001107, Virtual Entity of Relevant Acronyms, Jargon File, The Free On-line Dictionary of Computing, Easton's 1897 Bilble dictionary, Hitchcock's Bible Names dictionary, Bouvier's Law dictionary, The Devil's dictionary, CIAWorld Factbook 2002, U.S Gazetteer, U.S Gazetteer counties, U.S Gazetteer Places,* y *U.S Gazetteer Zip Code Tabulation areas.*

#### Tesauros

En las ciencias de la información, un tesauro es una lista de términos para representar conceptos o temas con descripciones y propiedades para facilitar el acceso y su manejo por parte de usuarios y sistemas de información. Para hacer polivalente y funcional este tipo de representaciones se suelen relacionar los términos entre sí con diferentes tipos de propiedades siendo habitual las relaciones de jerarquía, equivalencia o asociativas como sinónimos o antónimos.

Los tesauros utilizados en este campo son:

[9] http://www.dict.org

- WordNet[10] es el proyecto estrella por ser utilizado ampliamente en numerosos trabajos científicos relacionados con el tratamiento de lenguajes. WordNet es una gran base léxica de palabras en inglés. Nombres, verbos, adjetivos y adverbios son agrupados en conjuntos de sinónimos llamados *synsets*, donde cada uno dispone de su propia acepción. Los *synsets* están interrelacionados por relaciones léxicas y conceptualmente semánticas. El resultado es una red de conceptos y palabras relacionados por significado. Es gratuito y disponible públicamente para su descarga. Dada su estructura WordNet es útil para computaciones lingüísticas y para las técnicas de procesamiento del lenguaje natural. Existen una multitud de proyectos relacionados con WordNet que pueden ser consultados desde su propia web[11].
  La principal relación entre palabras en WordNet es la sinonimia. Los sinónimos son agrupados en *synsets*. WordNet contiene 117000 *synsets*, que son relacionados mediante relaciones estructurales de subordinación (llamadas relaciones *ISA*), relaciones de tipo (nombres comunes o individuales) y relaciones de meronimia (de composición). Las relaciones jerárquicas representan el 80 % de las relaciones entre diferentes *synsets*. En WordNet cada palabra es etiquetada según su función (conocido como *POS*): verbo, nombre, adjetivo y adverbio. Cada *synset* tiene una definición formada por un conjunto de palabras denominado *gloss*.
  La búsqueda de la palabra `art` en WordNet genera los resultados parcialmente representados en la figura 3.3.

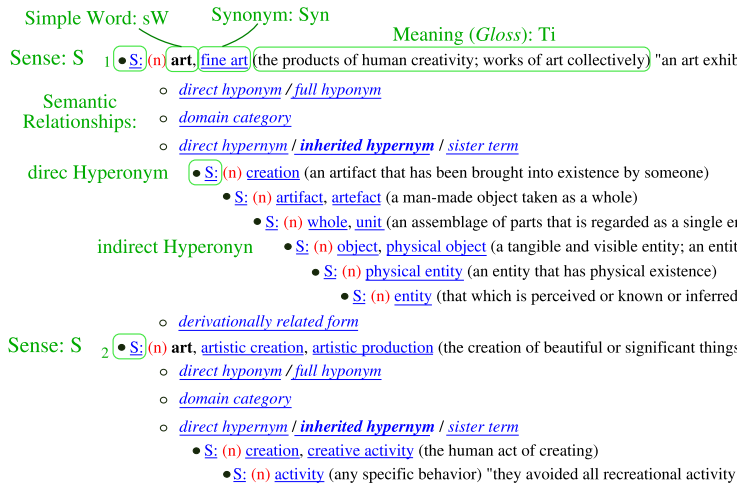


Figura 3.3: Información del concepto `art` visualizado por WordNet

[10] http://wordnet.princeton.edu/
[11] http://wordnet.princeton.edu/wordnet/related-projects/

- *Roget's thesaurus* es uno de los principales tesauros en inglés. Fue creado por Dr. Peter Mark Roget en 1805 y publicado en 1852. Esta compuesto de seis clases primarias. Cada clase está compuesta de múltiples divisiones y secciones que pueden ser conceptualizadas en una taxonomía. Dada una palabra obtenemos una serie de categorías. Estas categorías no son exactamente sinónimos pero pueden ser vistas como anotaciones del significado del concepto. El esquema diseñado por Roget de clases y subdivisiones está basado en el trabajo de Leibniz [76].
  La última versión data de 1987 publicada por Pearson Education y a la que pertenecen los derechos. La última versión gratuita es de 1911. Existen algunas herramientas como: The Open Roget's Project[12], The ARTFL Project para consultas online[13].
  Incluimos un ejemplo de algunas categorías de la palabra `meaning`, proporcionadas por el *Roget's thesaurus*, son: *uncertainty, incredulity, absurdity, [idea to be conveyed.] meaning [thing signified.], [absence of meaning.] unmeaningness, intelligibility, unintelligibility, [having a double sense] equivocalness, interpretation, misinterpretation, manifestation, latency, implication, untruth, obscurity, intention* y *loss*.
- *Moby Thesaurus* es el tesauro del proyecto Moby creado por Grady Ward[14]. El tesauro fue publicado en 1996 y el resto de paquetes del proyecto en 2007. El proyecto Moby consta de 5 diccionarios en diferentes lenguas (alemán, español, francés, italiano y japonés), de un diccionario de pronunciación del inglés, la obra integra de Shakespeare, un listado de palabras según funciones: acrónimos, palabras más usadas por USENET, nombres propios comunes, etc., el tesauro y un listado de las funciones de una palabra incluyendo preposiciones, artículos definidos e indefinidos, entre otras categorías.
  Concretamente el tesauro está estructurado de manera similar a *Roget's thesaurus* conteniendo más de 30000 palabras claves (clasificaciones) y más de 2.5 millones de sinónimos y términos relacionados. El formato de la información es ASCII para facilitar la adopción de cualquier herramienta. Cada palabra es seguida por una lista de palabras relacionadas separadas por coma y listadas alfabéticamente.
  Acorde a este tesauro, a la palabra `frill` le corresponden los siguientes términos: *frill, addition, adornment, amenity, beading, beauties, bedizenment, binding, bonus, bordering, bordure, bravery, chiffon, clinquant, colors, colors of rhetoric, crease, creasing, crimp, crisp, decoration, dog-ear, double, double over, doubling, duplication, duplication of effort,...*

La diferencia entre *Roget's thesaurus* y WordNet radica en dos hechos evidentes: la antigüedad del primero y la estructura del segundo. La clasificaciones de *Roget's thesaurus* facilitan la creación de estructuras basadas en

[12] http://rogets.site.uottawa.ca/
[13] http://artfl-project.uchicago.edu/
[14] http://icon.shef.ac.uk/Moby/

perfiles de palabras o en vectores, además cada categoría está identificada inequívocamente lo que facilita la estructura de métricas. Este tipo de construcciones en WordNet requieren un coste adicional durante el recorrido de los diferentes enlaces en la red de términos. Por otro lado, *Moby Thesaurus* ofrece un número mayor de palabras por lo que en ciertas operaciones no ofrece una precisión adecuada y además, hay que destacar su reciente aparición.

En numerosos trabajos relacionados con el mapeado de ontologías, WordNet ofrece un mayor repertorio de criterios dada la variedad de relaciones en ambas estructuras en comparación a *Roget's thesaurus*.

#### Ontologías de amplio dominio

Una ontología de amplio dominio es una ontología con conceptos generales de múltiples dominios de conocimiento. La principal función de una ontología de amplio dominio consiste en proporcionar mecanismos de interoperabilidad a un gran número de ontologías. Las categorizaciones de los diferentes niveles corresponden a concepciones filosóficas o naturales, aunque hay alguna de ellas que son particulares a la medicina. Podemos destacar ontologías como: DOLCE[15], SUMO[16], OpenCyc[17], BFO[18], COSMO[19], PROTON[20], entre otras.

### 3.3.5 Medidas estructurales con recursos externos

WordNet es de los recursos más empleado en el mapeado de ontologías. [16] recopilan algunas de las heurísticas utilizadas en múltiples trabajos, relacionados más estrechamente con técnicas de NLP pero permiten establecer el posible repertorio de medidas disponibles gracias a WordNet. Para entenderlas hay que definir una serie de conceptos básicos:

- La *longitud*, $len(c_i, c_j)$, es el camino más corto -el número de niveles jerárquicos- para llegar a relacionar dos *synsets*.
- La *profundidad* de un nodo es la longitud desde el concepto hasta la raíz, $depth(c_i) = len(root, c_i)$.
- El más específico subsumidor de dos conceptos dados, la mínima clase padre entre ambos, $lso(c_1, c_2)$.
- Dada cualquier fórmula de similitud relacional entre dos conceptos, $rel(c_1, c_2)$, la relación entre dos palabras, $rel(w_1, w_2)$, puede ser calculada:

$$rel(w_1, w_2) = max_{c_1 \in S(w_1), c_2 \in S(w_2)}[rel(w_1, w_2)] \tag{3.9}$$

[15] http://www.loa-cnr.it/DOLCE.html
[16] http://www.ontologyportal.org/
[17] http://www.opencyc.org/
[18] http://www.ifomis.uni-saarland.de/bfo/
[19] http://www.micra.com/
[20] http://proton.semanticweb.org/

donde $S(w_i)$ es "el conjunto de conceptos en la taxonomía que son acepciones de $w_i$". Es decir, la similitud de dos palabras es igual a la mayor relación de similitud que guarden sus conceptos a los que definan.

Los autores de [16] enumeran una serie de trabajos donde se ha experimentado con las siguientes ideas.

Una de las medidas básicas es calcular la longitud del camino entre dos conceptos: "A menor camino mayor es su similitud" [129]. Detrás de la idea de Resnik está la intuición de la cantidad de información que comparten en común:

$$sim_R(c_1, c_2) = -logp(lso(c_1, c_2)) \tag{3.10}$$

Donde $p$ es una función monótona según la ascensión en la taxonomía, como consecuencia a mayor altura en la posición del subsumidor menor será su similitud.

$$p(c) = \frac{\sum_{w \in W(c)} count(w)}{N} \tag{3.11}$$

Donde $W(c)$ es el conjunto de palabras subsumidas por el concepto $c$ y $N$ es el número total de palabras presentes en WordNet.

Otros autores, [56], definieron dos tipos de grado de similitud: fuerte y media. Así, dos conceptos tienen una relación fuerte si tienen un *synset* asociado o si sus respectivos *synsets* están conectados por una relación de antonimia o si uno forma parte del otro en una palabra compuesta. Por otro lado, mantienen una relación de grado medio si existe un camino *accesible* entre ellos. Un camino accesible se define como un camino de una longitud inferior a 5 y se adapta a alguno de los 8 patrones que definen. La idea subyacente son los cambios de dirección que sigue el camino dentro de la jerarquía. Finalmente proponen la siguiente fórmula para determinar la similitud entre dos conceptos:

$$rel_H S(c_1, c_2) = C - len(c_1, c_2) - k * turns(c_1, c_2) \tag{3.12}$$

Donde $C$ y $k$ son constantes prácticas (C=8 y k=1) y $turns(c_1, c_2)$ es el número de veces que el camino entre $c_1$ y $c_2$ cambia de dirección.

Hay otro tipo de técnicas basadas en escalar la distancia entre dos conceptos. En este sentido podemos mencionar los trabajos de Sussna [152] basados en la idea de que conceptos hermanos aparecen más cercanos los unos a los otros a medida que se asciende en la taxonomía. Asigna una ponderación, $r$, a cada relación de hiperonimia, hiponimia, holonimia, y meronimia entre los rangos $min_r = 1$ y $max_r = 2^4$. El peso de cada eje -relación de cada tipo $r$ de un nodo $c_1$- se reduce por un factor que depende del número de ejes, $edges_r$ del mismo tipo, contabilizados desde $c_1$:

$$wt(c_1 \rightarrow_r) = max_r - \frac{max_r - min_r}{edges_r(c_1)} \tag{3.13}$$

De esta manera la distancia entre dos nodos es la media de los pesos en cada dirección de los ejes, escalados por la profundidad de los nodos:

$$dist(c_1, c_2) = \frac{wt(c_1 \rightarrow_r) + wt(c_2 \rightarrow_{r'})}{2 * max\{depth(c_1), depth(c_2)\}} \tag{3.14}$$

Wu y Palmer [164] definieron la siguiente métrica a la que llamaron similitud conceptual:

$$sim_{WP}(c_1, c_2) = \frac{2 * depth(lso)}{len(c_1, lso) + len(c_2, lso) + 2 * depth(lso)} \tag{3.15}$$

donde la abreviatura $lso$ corresponde a $lso(c_1, c_2)$.

Leacok y Chodorow [86] propusieron la siguiente fórmula:

$$sim_{LC}(c_1, c_2) = -log \frac{len(c_1, c_2)}{2 * max_{c \in WordNet} depth(c)} \tag{3.16}$$

Algunas de estas técnicas son nombradas y utilizadas en las propuestas de OM. Otras sufren ciertos cambios para adaptar relaciones al repertorio de constructores de OWL. Otro grupo de heurísticas han considerado comparar la estructura de WordNet con las ontologías ya que WordNet puede considerarse como una taxonomía. En algunos casos se nombra a WordNet como una ontología lo cual es incorrecto por la función y el lenguaje utilizado en su definición. WordNet es un tesauros y un tesauros no es una ontología, pero sí al revés. Antes de finalizar este capítulo haremos un análisis de cada una de las técnicas usadas en las propuestas presentadas.

### 3.3.6 Combinacionales

Como se ha comentado en la figura 3.1, las técnicas presentadas anteriormente pueden ser consideradas como bloques simples. Para mejorar la precisión de los resultados se requiere de una estrategia más compleja que combine los diferentes bloques simples.

Algunas de las técnicas son aplicadas secuencialmente o en paralelo con el posible refinamiento iterativo.

Una de las estrategias más utilizada es la *suma ponderada*. Diferentes técnicas/bloques simples son aplicadas y sus resultados son ponderados generalmente con pesos fijados de manera experimental para proveer una única medida de similitud. Por ejemplo, podemos definir una función de similitud por la distancia de Jaro-Winkler y por la distancia de Leacok y Chodorow con dos pesos ($\alpha$ y $\beta$) relativos a lo que consideremos oportuno:

$$sim_{ej}(c_1, c_2) = \alpha sim_{JW}(c_1, c_2) + \beta sim_{LC}(c_1, c_2) \tag{3.17}$$

En algunos casos la superación de un umbral por parte de una técnica habilita la aplicabilidad de otra que fija el resultado. Este umbral puede estar fijado experimentalmente o puede ser dinámico bajo una determinada función, como por ejemplo, una función sigmoide.

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \tag{3.18}$$

En otros casos, los resultados de diferentes bloques pueden combinarse en un modelo de vectores permitiendo la aplicabilidad de cálculos como la distancia de coseno o utilizar medidas combinacionales como el coeficiente de Jaccard o Dice, o el coeficiente de correlación de Pearson, o de Spearman, etc.

El repertorio de opciones es específico de cada una de las propuestas por lo que resulta difícil englobarlas todas.

## 3.4 Evaluación

Debido a la dificultad de formalizar el problema, el rendimiento de las técnicas de mapeado ha de evaluarse experimentalmente. Hay tres tipos de métodos: mediante la comparación con juicios personales, mediante métricas o funciones que permiten compararse entre ellas, y su aplicabilidad en un entorno de trabajo donde la productividad de los resultados influya en la productividad de la aplicación.

### 3.4.1 Medidas de rendimiento

Las medidas más comunes tienen como origen la medidas clásicas de la recuperación de información (*Information Retrieval*), estas son: *recall*, precisión (*accuracy*) y fallos (*fallout*). *Recall* ($R$) se define como el porcentaje de elementos clasificados -o acertados- bajo una cierta categoría respecto al total de elementos pertenecientes a esa categoría. Precisión ($P$) es el porcentaje de elementos clasificados correctamente entre el total de elementos que fueron asignados a esa categoría. Fallos es la proporción de asignaciones incorrectas dado el número de clases incorrectas que el sistema puede generar. Idealmente, *recall* y precisión han de valer 1, y el porcentaje de fallos 0 [43; 49; 110].

$$R = a/(a + c) \tag{3.19}$$

$$P = a/(a + b) \tag{3.20}$$

$$Fal = b/(b + d) \tag{3.21}$$

| | Experto: Sí | Experto: No | |
|---|---|---|---|
| Sistema: Sí | $a$ | $b$ | $a + b = k$ |
| Sistema: No | $c$ | $d$ | $c + d = n - k$ |
| | $a + c = r$ | $b + d = n - r$ | $a + b + c + d = n$ |

donde, $n$ es el número de objetos clasificados, $k$ es el número de objetos clasificados por el sistema y $r$ es el número de objetos clasificados por el experto.

A la hora de comparar diferentes clasificadores o técnicas es deseable tener una única medida. La *F-measure* es usada como métrica combinando *recall* y precisión:

$$F = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \tag{3.22}$$

donde $P$ es la precisión, $R$ es el *recall* y $\beta$ es un factor que indica la importancia relativa de *recall* y de la precisión. Cuando $\beta$ es igual a 1, ambas medidas tienen la misma importancia y esta métrica es conocida como $F_1$-*measure*: $\frac{2}{1/R+1/P}$.

Existen otras medidas como la exactitud $(a + d/n)$ y el ratio de error $(b + c/n)$.

### 3.4.2 *Benchmarks*

*Ontology Alignment Evaluation Initiative* (OAEI)[21] es una iniciativa internacional que organiza la evaluación de los sistemas de mapeado mediante el consenso y la coordinación. Sus principales objetivos son: la evaluación de los puntos débiles y fuertes de diferentes propuestas, la comparación del rendimiento, aumentar la comunicación entre los desarrolladores de algoritmos, mejorar las técnicas de evaluación y, en definitiva, mejorar la calidad de las propuestas. En pocas palabras, el principal objetivo de OAEI es la comparación de sistemas teniendo en cuenta su funcionamiento interno para poder extraer conclusiones sobre sus estrategias en función de la calidad de los resultados. Desde 2004 se han ido realizando eventos anuales con el objetivo de difundir y comunicar a los desarrolladores, sus sistemas y sus correspondientes resultados.

La metodología de la campaña [39] consiste en la presentación de los casos a los participantes. Posteriormente, se presenta el entorno de evaluación y se permite el inicio de los experimentos. Finalmente, se recolecta toda la información de los diferentes escenarios y se realiza un informe general de la campaña.

La evaluación está formada por una serie de guías y casos de estudios. Las guías tienen como objetivo establecer un marco de trabajo común para todos los sistemas dentro de un grupo de análisis en común. (i) La primera prueba está basada en un ontología particular con un dominio centrado en definir la bibliografía y otro conjunto alternativo de ontologías del mismo dominio donde los alineamientos son dados. (ii) La segunda consiste en evaluar la expresividad mediante dos casos. El primero de ellos son dos representaciones para describir la anatomía humana: *Adult Mouse Anatomy* y *NCI Thesarurus*. El segundo de ellos, es un conjunto de ontologías para describir conferencias. Los resultados son evaluados automáticamente mediante alineamientos de referencias. Algunos casos son evaluados manualmente. (iii) El tercer estudio

[21] http://oaei.ontologymatching.org/

consiste en la comparación de directorios reales, como por ejemplo: el directorio de Yahoo. (iv) El cuarto, evalúa la comparación de individuales en RDF de fuentes diferentes pero que describen las mismas entidades del mundo real. Se divide en dos casos: *Data Interlinking* (DI) con el objetivo de reconstruir enlaces entre las instancias RDF y, el segundo caso llamado *OWL data track* (IIMB & PR) los individuales son representados en OWL en dos representaciones IIMB y PR.

Los resultados para ser evaluados han de representarse mediante el formato de *Alignment API*[22]. Las medidas de comparación son las presentadas anteriormente: *recall*, precisión y *F-measure*.

OAEI ha marcado un antes y un después recopilando y fomentando la evaluación de sistemas. Antes de 2004, los sistemas presentaban evaluaciones específicas basadas en directorios, ontologías particulares o herramientas específicas de edición [83; 105; 119; 151]. Actualmente, existen otras propuestas específicas de evaluación pero están orientadas a dominios medicinales basados en UMLS [83; 85].

Actualmente, OAEI sigue marcando las pautas para la referenciación de algoritmos y puesta a punto de pruebas, pero en los dos últimos años (2010 y 2011) el proyecto *Semantic Evaluation At Large Scale* (SEALS)[23], del 7º programa marco, ha estado desarrollando la infraestructura de referencia para facilitar la evaluación formal. Esto permite tanto a campañas de gran escala, las realizadas en OAEI, como también a evaluaciones puntuales a nivel personal o organizativas realizar tales pruebas. Dentro de este proyecto participan el laboratorio INRIA donde pertenece [39]. Esto ha hecho que algunas pruebas de OAEI se realicen bajo esta infraestructura facilitando la aplicabilidad de pruebas y análisis de resultados de los diferentes participantes de la campaña.

### 3.4.3 Otros casos

Existe algunas propuestas que han basado su evaluación en la comparación de dos catálogos -taxonomías- referentes a información académica de la Universidad de Cornell[24] y de la Universidad de Washington[25]. Los cursos son organizados en escuelas y colegios, con sus respectivos departamentos y centros.

Algunas utilizan directorios de compañías conocidas para evaluar este resultado: Yahoo[26], y The Standard[27], que describen la situación económica de las compañías. Las compañías son organizadas en sectores y las industrias,

[22] http://alignapi.gforge.inria.fr/
[23] http://www.seals-project.eu/
[24] http://courses.cornell.edu/
[25] http://www.washington.edu/students/gencat/
[26] http://dir.yahoo.com/
[27] http://www.microsoft.com/biztalk/en/us/default.aspx

en cada sector. Google ya no ofrece servicio de directorio, ahora remite a OpenDirectory[28] y TheStandard fue adquirirá por Microsoft.

## 3.5 Representación de alineamientos

Con el requisito de representar los alineamientos mediante el formato definido en Alingment API en las campañas de OAEI, éste ha sido el formato de representación por referencia. Desde su primera versión [38] hasta la última [31] presenta los mismos objetivos. (i) Desde el punto de vista semántico, la posibilidad de encontrar y reusar alineamientos dinámicamente. (ii) Desde el punto de vista de la ingeniería del software, la posibilidad de intercambiar resultados entre diferentes programas. (iii) Desde el punto de vista de la ingeniería de ontologías y gestión, la integración con el ciclo de vida de una ontología. Alingment API contiene implementado un conjunto de repertorio de operaciones simples y complejas explicadas anteriormente. A nivel de representación, los resultados inicialmente fueron representados en RDF, involucrando los elementos y el tipo de relación entre ellos (ver código 3.4).

```
<?xml version=1.0 encoding=utf-8 standalone=no>
<!DOCTYPE rdf:RDF SYSTEM "align.dtd">
<rdf:RDF
  xmlns=http://knowledgeweb.semanticweb.org/heterogeneity/alignment
  xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:xsd=http://www.w3.org/2001/XMLSchema#>
<Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
  <onto1>http://www.ontology1</onto1>
  <onto2>http://www.ontology2</onto2>
  <map>
    <Cell>
      <entity1 rdf:resource=http://www.ontology1#reviewedarticle/>
      <entity2 rdf:resource=http://www.ontology2#article/>
      <measure rdf:datatype=&xsd;float>0.6363636363636364</measure>
      <relation>=</relation>
    </Cell>
  </map>
  <map>
    <Cell>
      <entity1 rdf:resource=http://www.ontology1#journalarticle/>
      <entity2 rdf:resource=http://www.ontology2#journalarticle/>
      <measure rdf:datatype=&xsd:float>1.0</measure>
      <relation>=</relation>
    </Cell>
  </map>
</Alignment>
</rdf:RDF>
```

Figura 3.4: Ejemplo de representación en formato *Alignment API* v.3

[28] http://www.dmoz.org/

En la última versión la representación se realiza mediante *Expressive and Declarative Ontology Alignment Language* (EDOAL)[29]. La principal novedad de EDOAL es la capacidad de integrar en una relación un conjunto de descripciones y no solo una identidad. Como por ejemplo, $\{Wine \wedge hasTerroir \cdot locatedIn = Aquitaine\} \geq Bordeaux$ donde la relación $\geq$ es el alineamiento entre ambas fuentes descriptivas (ver código 3.5).

```
<al:entity1>
 <ed:Class rdf:about="Bordeaux"/>
</al:entity1>
<al:entity2>
 <ed:Class>
  <ed:and rdf:parseType="Collection">
   <ed:Class rdf:about="Wine"/>
   <ed:AttributeValueRestriction>
    <ed:onAttribute>
     <ed:Relation>
      <ed:compose rdf:parseType="Collection">
       <ed:Relation rdf:about="hasTerroir"/>
       <ed:Relation rdf:about="locatedIn"/>
      </ed:compose>
     </ed:Relation>
    </ed:onAttribute>
    <ed:comparator rdf:resource="&xsd:equals"/>
    <ed:value>
     <ed:Instance rdf:about="Aquitaine"/>
    </ed:value>
   </ed:AttributeValueRestriction>
  </ed:and>
 </ed:Class>
</al:entity2>
<al:measure rdf:datatype=&xsd:float> 1. </al:measure>
<al:relation> SubsumedBy </al:relation>
</al:Cell>
```

Figura 3.5: Ejemplo de representación en formato *Alignment API* v4.

En [127] los autores definen una ontología llamada SCROL (*Semantic Conflict Resolution Ontology*) que es usada para identificar y resolver conflictos semánticos entre bases de datos heterogéneas. En su trabajo mencionan dos propuestas más como son Clio [109] y Cupid [97] y su diferencia con ellas pues defienden que en estas propuestas el usuario ha de estimar como son de iguales y de diferentes para resolver manualmente estás diferencias. En cambio, en SCROL estas diferencias están bien definidas y se intenta automatizar la integración de estas diferencias, con la misma opinión de los autores [75]. La idea de SCROL surge unos años antes de que se definiera OWL en 2004. Mucho de los constructores definidos en SCROL son similares a los de OWL. Hay conceptos, instancias, relaciones de jerarquía entre conceptos, y relaciones de hermandad (*sibling relationships*), relaciones de disyunción, relaciones de contemporaneidad *(peer relationships*), relaciones de *parte de*, y relación *is-a*.

[29] http://alignapi.gforge.inria.fr/edoal.html

Como SCROL está enfocado a bases de datos incorpora notaciones de cardinalidad. El lenguaje de representación de alineamientos mediante SCROL no está clarificado en el artículo. En las conclusiones indican que estaban explorando la posibilidad de incorporar XML. Los resultados tienen forma de tupla con los elementos involucrados y el tipo de relación entre ellos. Como caso de alineamiento, presentan estas relaciones `(Square Meter, Acre, 1-1)` o `(City, County, part-of)`.

SCROL puede considerarse un lenguaje innovador para su época por la consideración del contexto, la transparencia de resultados, y el uso del tipo de relación en los alineamientos.

## 3.6 Propuestas

A continuación exponemos un listado de aquellas propuestas que han sido especialmente significativas o han participado en las campañas propuestas por OAEI para su evaluación. Hemos optado por seleccionar aquellas soluciones de esta última década aproximadamente. La mayoría de estos trabajos citan trabajos anteriores y la amplia mayoría se encuentran en el libro de [41] o bien, en artículos del estado del arte [22; 40; 51; 140].

Estos trabajos están ordenados por año de aparición. No se ha pretendido en ningún momento comparar los resultados obtenidos: ellos mismos han pasado por lo que han considerado un adecuado sistema de evaluación. En la siguiente sección veremos a nivel analítico cada una de las métricas, combinaciones, evaluaciones, representaciones y hechos característicos de cada una de ellas con respecto al mapeado de ontologías.

### 3.6.1 COMA

COMA [53] es una estrategia híbrida que combina diferentes medidas para relacionar esquemas XML y relacionales. El sistema da la posibilidad de que un usuario contraste los resultados y se ha implementado un gestor de base de alineamientos para futuras reutilizaciones. A nivel léxico emplea diferentes medidas: $q$-gram, similitud por sinonimia, uso de la totalidad de cada elemento, etc. A nivel estructural, comprueba el número de descendientes y hojas entre los diferentes elementos como valor inherente de similitud. Incorpora diferentes estrategias de combinación donde se aplican medias, medias ponderadas, selección por rango y el coeficiente de DICE. Para la evaluación usaron 5 esquemas XML donde fueron probando cada una de las diferentes combinaciones. En esta primera versión, vemos evidente que el objetivo es el esquema independientemente del lenguaje. COMA++ [5] es una extensión de COMA con la compatibilidad con el estándar OWL-lite, al mismo tiempo admitiendo SQL y W3C XSD, de definición de esquemas.

### 3.6.2 GLUE

GLUE [35] es una solución basada en la distribución conjunta junto con dos tipos de heurística para complementar los resultados de la primera. El algoritmo calcula la probabilidad conjunta de dos conceptos en sus respectivas taxonomías, mediante la similitud Jaccard 3.3.1. Así, determina aquellos individuales pertenecientes a la unión de ambas clases.

Las dos técnicas heurísticas se basan en la *proximidad de vecindad* producida por la *relajación del etiquetado* y en una serie de restricciones. La *relajación del etiquetado* en clases vecinas suele ser común en taxonomías. Se produce cuando el nombre de las subclases contiene el nombre de la clase padre, por ejemplo, la clase padre: `Wine` y dos subclases: `WhiteWine` y `RedWine` comparten el término *Wine*. Las restricciones sobre esta heurística se basan en casos específicos como: "dos clases serán similares si sus parientes son similares al menos un %", "si todos las subclases de un nodo X coinciden con todas las subclases de un nodo Y entonces X e Y son similares". Además, definen una serie de restricciones específicas a su dominio, por lo que este tipo de restricciones afectan directamente a la evaluación.

La evaluación se realiza en tres pares de taxonomías: Cornell-Washington, Cornell-Washington(2), Standard.com-Yahoo.com. Las dos primeras evaluaciones corresponden a dos catálogos del curso de las universidades de Cornell y de Washington. En el segundo caso, el catálogo es más extenso. Todos los catálogos han pasado un proceso de filtrado descartando conceptos con menos de 5 individuales debido a la imposibilidad de aplicar el algoritmo. La precisión se ha calculado mediante la comparación de los resultados realizados por personas.

Los autores comentan que aplican técnicas de *machine learning* pero no explican, ni definen el tipo de aprendizaje, ni los datos históricos utilizados para tal fin, por tanto se descarta esta solución como un sistema basado en *machine learning* ateniéndonos a la definición de un sistema de aprendizaje [2]. En la pág. 10 de [35] definen el conjunto de "aprendedores" basados en el contenido, el nombre y la meta información de los conceptos. Las tres medidas resultantes son sumas ponderadas, sin recurrir a datos históricos. A falta de explicación, los alineamientos no son representados mediante ningún lenguaje específico. Además, podríamos preguntarnos cuál es el peso sobre los resultados del porcentaje de clases descartadas por no poseer 5 individuales y las 4 restricciones dependientes del dominio.

### 3.6.3 S-Match

S-Match [48; 123] es un algoritmo caracterizado por basarse en la similitud de clases en vez de etiquetas. Uno de los primeros en definir tipos de relaciones como resultado del alineamiento en vez de un valor numérico y en utilizar el significado de las clases para mapear. Su solución se basa en dos ideas: "concept of a label" y "concept at a node". El primero hace referencia al

conjunto de instancias que se podrían englobar en esa etiqueta y la segunda idea hace referencia al conjunto de instancias que se podrían agrupar a un cierto nodo con referencia a una etiqueta y a una cierta posición dentro del grafo. S-Match realiza cuatro pasos: el primero determina que concepto se haya detrás de cada etiqueta mediante el uso de WordNet. Las etiquetas compuestas se contemplan como la unión de ambas excepciones, como por ejemplo el concepto $C_{canalhistoria} = C_{canal} \sqcap C_{historia}$. El segundo paso consiste en determinar la acepción en función de la posición en el árbol, desde un punto de vista jerárquico para la clasificación y subsunción de palabras. Tercer paso, computa la similitud en base al etiquetado de las clases aplicando una serie de comparadores en el siguiente orden: prefijos, sufijos, *edit distance*, $q$-gram, Corpus textual, WordNet, distancia jerárquica, el *gloss* de WordNet, la versión extendida del *gloss* de WordNet, comparación del *gloss*, comparación extendida del *gloss*, comparación semántica del *gloss* y comparación semántica extendida del *gloss*. Los cinco primeros son basados en comparación de caracteres. Los dos siguientes en comparación de la acepción, y el resto son basados en el *gloss*. El último paso consiste en determinar las relaciones entre los conceptos de nodos. La correcta elección de relaciones entre conceptos la han basado en un problema de satisfacción booleana (SAT). El tipo de alineamiento semántico se convierte en una tabla de verdad con el objetivo de encontrar el vacío o una contradicción. Además, para simplificar la magnitud de relaciones han incorporado varias técnicas de simplificación de proposiciones.

Los alineamientos se definen en base a cuatro tipos de relaciones: equivalencia (=), subsunción (más general y menos general), sin correspondencia ($\perp$) y con solapamiento ($\sqcap$). Este último tipo no lo han implementado ya que WordNet no provee tal categoría. Los alineamientos son representados mediante XML.

La evaluación consistió en 6 experimentos: la información de los cursos de la universidad de Cornell y de Washington, información empresarial de dos esquemas CIDX y Excel; y tres tipos de correspondencias entre directorios: Google, Yahoo y Looksmart. Todos los experimentos y las fuentes se encuentran en la web[30].

Entre la primera versión en 2004 y la última versión del 2010 se aprecia la evolución del trabajo al igual que el tópico ha ido evolucionando. Podemos destacar cambios significativos en la evaluación, en las técnicas de SAT y optimización de proposiciones y en la arquitectura del sistema.

### 3.6.4 OLA

OLA [42] es una solución específica para OWL-Lite. Este tipo de representaciones poseen un mapeado con un tipo especial de grafos y, por tanto, el problema se reduce a encontrar similitudes entre nodos de ambos grafos. La

[30] http://disi.unitn.it/∼knowdive/description_SMatch_experiments.php

similitud resultante, un valor entre 0 y 1, es la suma ponderada de las diferentes similitudes consideradas. Estas son propiedades *datatype* y *object*, la firma de la propiedad, tipo, el valor de los literales y la etiqueta.

La propuesta tal como comentan los autores se centran en la descripción estructural de los elementos, es decir, mismas descripciones hacen similares a esos conceptos. La correcta justificación del cálculo de las diferentes similitudes está vagamente explicado. No existe una evaluación o comparación con otras iniciativas.

### 3.6.5 Falcon-AO

Falcon-AO [67] es una composición de dos algoritmos de mapeado: uno léxico (LMO) y el otro estructural (GMO). El algoritmo lingüístico combina a su vez otras dos medidas: la distancia de edición de la etiqueta y una distancia heurística basada en la aparición de términos acontecidos en los comentarios y en las diferentes etiquetas que pueda presentar las clases vecinas de la clase. Ambas medidas son unificadas mediante una suma ponderada por pesos experimentales. La medida estructural se basa en un cálculo propio de anteriores trabajos en comparar grafos. La medida se basa en comparar tripletas con información estructural similar: sujeto, predicado y objeto. Según los autores, para que el proceso de similitud estructural sea más preciso se utiliza la medida de similitud léxica mediante la fórmula de similitud de coseno. La evaluación se realiza mediante el conjunto de datos de OAEI 2005.

Acorde a los autores, Falcon-AO obtiene buenos resultados cuando ambas ontologías comparten muy pocos aspectos de similitud léxica pero son estructuralmente muy compatibles.

Falcon-AO [58] integra cuatro tipos de algoritmos (V-Doc, I-Sub, GMO y PBM) en vez de los dos iniciales. Los resultados son representados en RDF/XML e incorpora un repositorio de alineamientos con el objetivo de minimizar recursos en sucesivas comparaciones. Los algoritmos V-Doc y I-Sub son algoritmos basados en medidas léxicas; GMO, en similitudes estructurales y PBM divide en regiones por parecido estructural las ontologías, donde el resto de algoritmos trabajan de manera independiente. El algoritmo es evaluado mediante el conjunto de datos de OAEI 2007.

### 3.6.6 MoA

MoA [79] es un algoritmo basado en la idea del significado global y local de un concepto. Los autores centran sus esfuerzos en solventar dos casos "habituales" en el diseño de ontologías: el uso de múltiples palabras en el etiquetado de la clase y la propagación de etiquetas en las jerarquías. Generan una estructura para gestionar el significado de cada palabra mediante su identificación en WordNet. Las comparaciones se realizan mediante reglas donde se comprueba que las palabras implicadas tienen alguna relación de equivalencia o de jerarquía con palabras ubicadas en WordNet. El resultado no es un alineamiento

sino la reestructuración de la fusión de ambas ontologías. La evaluación se base en tres análisis de casos específicos: una comparación de ontologías de alquiler de coches y reserva de vuelos, dos ontologías de sendas organizaciones y dos ontologías de transporte, una desarrollada por Teknowledge Corporation y la otra por CYC. La evaluación se basa en alineamientos establecidos por personas.

### 3.6.7 SAMBO

SAMBO [84] es un sistema para mapear y combinar ontologías biomédicas. El uso de ontologías biomédicas ha crecido en la última década, tal ha sido su importancia que se han creado organismos para su gestión y calidad como Gene Ontology Consortium[31] en 1998. Este tipo de ontologías se caracteriza por una terminología específica, por un número elevado de elementos y por ser, básicamente, taxonomías. SAMBO es un sistema que combina diferentes estrategias mediante una suma ponderada. La elección de las estrategias y los pesos recae en el usuario. Las estrategias se basan en: (i) Un mapeado de terminologías, son dos técnicas léxicas como son $q$-gram y distancia de edición. Para aumentar la probabilidad de acierto cada palabra es preprocesada por un analizador léxico, en este caso Porter, junto con los posibles sinónimos y superclases en WordNet. (ii) Un mapeado estructural, es un algoritmo iterativo basado en las relaciones de `is-a` (rdf:type) y `part-of` (owl:subclassOf) en las jerarquías. Este algoritmo necesita de alineamientos previos para cotejar su viabilidad calculando la distancias en la jerarquía mediante las dos anteriores relaciones. (iii) Con conocimiento del dominio, los autores en este caso utilizan el Metatesauro de *Unified Medical Language Systems* (UMLS)[32] el cual está organizado mediante conceptos que comparten sinónimos y significado. La similitud de dos términos se calcula mediante la similitud en UMLS. (iv) Un sistema de autoaprendizaje que consiste en la clasificación del concepto en una serie de resúmenes de publicaciones médicas. Dos conceptos son similares si comparten los mismos resúmenes.

La evaluación se basa en ontologías especialmente creadas para tal fin mediante alineamientos proporcionados por expertos. Consideraron dos tipos de alineamientos basados en equivalencias y en relaciones de tipo (`rdf:type`). Además, los autores comparan su propuesta con tres herramientas: Protégé, PROMPT [118] y FOAM [36]. Los resultados de SAMBO son mejores o iguales a los del resto en las diferentes pruebas. En este estudio no está indicado el peso desempeñado por los recursos que manejan conocimiento sobre el dominio.

### 3.6.8 GeRoMeSuite

GeRoMe [77] es un metamodelo genérico con el objetivo de simplificar tareas al gestor del modelo tales como la integración, la evolución, el mapeado a nivel

[31] http://www.geneontology.org/GO.consortiumlist.shtml
[32] http://www.nlm.nih.gov/research/umls/

conceptual o al nivel lógico del diseño. A partir del metamodelo definido en GeRoMe, se desarrollo el algoritmo de mapeado llamado GeRoMeSuite [78]. Mediante GeRoMe, los elementos del modelo son descritos con una serie de roles. Dependiendo de la naturaleza del proceso ciertos roles se usan y el resto pasan desapercibidos. Es decir, la idea es disponer de diferentes visiones de un elemento. Por citar algunos de sus roles, ellos definen: asociación, unión, `is-a`, enumeración, atributo, literal, visible, referencia, clave principal, etc. Los autores definen la arquitectura y diferentes ejemplos de metamodelos usando los siguientes lenguajes de modelado: modelos relacionales, UML, XML y OWL. En GeRoMeSuite múltiples estrategias se pueden combinar. Los autores consideran dos tipos de estrategias: léxicas y estructurales. La medida léxica se basa en la comparación de etiquetas. A nivel estructural usan la medida *ChildrenMatcher* donde la similitud de los nodos se basa en la similitud de sus descendientes directos. Al ser un entorno de trabajo no definen claramente el resto de estrategias utilizadas. Cuando se producen conflictos entre los resultados parciales, la intervención manual se hace indispensable. Los resultados son valores entre 0 y 1.

### 3.6.9 AROMA

AROMA [32] es un algoritmo basado en el descubrimiento de reglas de asociación entre diferentes elementos. Las reglas de asociación determinan la subsunción entre dos elementos y se basan en la jerarquía de la estructura. Un trabajo parecido, basado en jerarquías, por parte de los mismos autores confirma la resolución de estructuras taxonómicas [33]. Para buscar estas correspondencias entre la estructura se utiliza a nivel léxico la medida JaroWinkler 3.3.1. La idea detrás de las reglas de asociación es que un elemento estará subsumido por otro elementos si su vocabulario y descendientes están englobados por el elemento subsumidor. La evaluación consta de dos pruebas: la primera mediante el catálogo de cursos de la Universidad de Washington y Cornell y la segunda en la iniciativa OAEI. Según los autores, los resultados en la primera son esperanzadores pues ambas son jerarquías de dominios afines.

### 3.6.10 LILY

LILY [159] está basado en la idea de dividir en subgrafos el modelo original y en ellos aplicar medidas de similitud léxica y estructural. Una vez descubiertos estos alineamientos se propagan a lo largo de la reconstrucción del modelo. Las técnicas de similitud aplicadas son obviadas por los autores que simplemente comentan la información considerada: etiquetado, comentarios, jerarquía, propiedades relacionadas, la firma de las propiedades y los individuales. Todos estos cálculos son sumados ponderadamente donde los pesos son fijados experimentalmente. La evaluación se llevó a cabo en OAEI 2007. Según los autores LILY presenta dificultadas a la hora de manejar ontologías de gran tamaño durante el proceso de extracción de los subgrafos.

### 3.6.11 SEMA

SEMA [145] combina algoritmos léxicos, estructurales y semánticos en una determinada secuencia para explotar los resultados de los algoritmos previos para encontrar alineamientos adicionales. Consta de 6 tipos diferentes de algoritmos. (i) El primer algoritmo es COCLU [153]. COCLU fue propuesto originalmente para encontrar similitudes tipográficas entre secuencias de caracteres sobre un alfabeto (ASCII o UTF). Agrupa las palabras en diferentes grupos (*clusters*) mediante el algoritmo *Huffman tree*. Después de la creación de los grupos, el algoritmo devuelve el grado de similitud entre dos palabras. En función de estos primeros resultados se aplican tres reglas para establecer los primeros alineamientos. Primera regla, un par de elementos son similares si la similitud entre su etiquetado, etiquetas o comentarios superan un cierto umbral. Segunda regla, un par de elementos se considerarán aptos para ser mapeados si su etiquetado, etiquetas o comentarios superan un cierto umbral. Tercera y última regla, si no se producen las anteriores dos reglas, el etiquetado de ambos elementos se sustituirá por sinónimos de WordNet, volviendo a aplicar las dos primeras reglas. (ii) El segundo algoritmo, se basa en encontrar "rasgos latentes". Consiste en transformar en vectores cada elemento explotando lo que denominan "elementos de vecindad: etiquetas, comentarios, instancias, propiedades, jerarquías, firmas de propiedades, etc. con respecto a la semántica de la especificación". El vector tiene la misma longitud que el total de elementos en ambas ontologías, y en cada entrada figura la "frecuencia de cada palabra en la vecindad del correspondiente elemento". Este algoritmo está adaptado aplicando el proceso *Latent Dirichlet Allocation* (LDA) [10]. La idea es disminuir la ambigüedad con la descripción de los elementos en casos de polisemia y sinonimia. (iii) El tercer algoritmo está basado en el modelo de vector, sin indicar los datos utilizados. (iv) El cuarto algoritmo tiene en cuenta las instancias de las clases. Si por encima del 10 % de las instancias coinciden, ambas clases son consideradas equivalentes. (v) El quinto, es un algoritmo estructural que tiene en cuenta todos los alineamientos anteriores. Si dos clases contienen al menos un par de superclases coincidentes o subclases entonces ambas se consideran coincidentes. (vi) El último algoritmo, al igual que el anterior, se basa en las propiedades. Si dos clases comparten propiedades sobre un mínimo de un 90 % equivalentes entonces se consideran coincidentes. Todos los porcentajes son experimentales. La evaluación se realizó en la campaña de OAEI 2007.

### 3.6.12 DSSim

DDSim [112] sigue un proceso basado en matrices de similitudes de todos los elementos de ambas ontologías. Cada matriz está determinado por un tipo diferente de "experto". La combinación de matrices se realiza mediante la teoría de evidencias de Dempster-Shafer [137], permite gestionar incertidumbre en valores numéricos. El artículo trata la selección de los valores matriciales en

función de esta teoría de evidencias por lo que no se detallan con suficiente rigurosidad las medidas elegidas. Usa WordNet y la similitud léxica Jaccard. La evaluación se realizó a cabo en sendas campañas de OAEI 2006 y 2007.

### 3.6.13 PRIOR+

PRIOR+ [100] utiliza información lingüística y estructural para solventar el problema del mapeado. Las similitudes se encuentran definidas en una versión previa del sistema: PRIOR [101]. Para establecer las comparaciones cada clase tiene asociado un perfil con la siguiente información: identificador, nombre, comentarios, "más otro tipo de información descriptiva". Cada concepto tendrá asociado una ponderación basada en el la frecuencia de aparición del termino y la frecuencia inversa del documento (TF/IDF). Cada perfil puede propagarse en la jerarquía disminuyendo a la mitad su peso en cada salto jerárquico. Con el perfil y su correspondiente peso se obtiene un modelo de vector, que define las matrices resultantes. En función de las matrices resultantes de las anteriores tres medidas, se calcula tres valores numéricos mediante el cálculo de su armonía ($h_k$). La armonía (ver ecuación 3.23) se calcula como el número de celdas con el mayor grado de similitud ($\#CMAX_{M_k}$) correspondiente a una fila/columna en la matriz $M_k$ por la división del menor tamaño de ambas ontologías ($E_{o_i}$ número de elementos de la ontología $O_i$). Este valor de armonía permite ponderar cada una de las matrices en su unión. La matriz resultante tiene también asociada una armonía, según la superación de un cierto umbral los alineamientos tendrán que ser fijados por un tercer proceso llamado *NN-based Constraint Satisfaction Solver*. Tanto PRIOR y PRIOR+ han pasado por las campañas de AEOI 2006 y 2007 respectivamente.

$$h_k = \frac{\#CMAX_{M_k}}{min(E_{o_1}, E_{o_2})} \tag{3.23}$$

### 3.6.14 SeSA

SeSA [3] no está diseñado específicamente para ontologías si no para modelos conceptuales, donde sí podemos englobar a las ontologías. Por tanto, SeSA es capaz de encontrar alineamientos en modelos relacionales, en modelos basados en UML y en ontologías. La idea consiste en convertir cada modelo en un grafo conceptual, donde los ejes representan propiedades de los modelos junto con una ponderación según el tipo de propiedad. Por tanto, el mapeado de un modelo simplemente es encontrar similitudes entre los nuevos grafos. El tipo de propiedades se basan en relaciones de pertenencia (*is-a*) y en el tipo de cardinalidad. En el caso de las ontologías, la cardinalidad de una propiedad funcional es 0 o 1. La evaluación se realizó mediante un conjunto de modelos definidos en *GeneExpress Data Management* (GXDM) [102].

### 3.6.15 TaxoMap

TaxoMap [19], los autores en esta segunda versión consideran una ontología como una relación taxonómica por el gran número de ontologías utilizadas para tal fin sin que haya una especificación adecuada de las clases, propiedades e instancias. Definen tres tipos de resultados por equivalencia, por jerarquía y por "semantically related". Estas últimas, cuando se desconoce el *tipo de relación*. La equivalencia se establece por similitud de etiquetado. La relación de subsunción mediante la comprobación de la inclusión de la etiqueta. La comparación de nombres se realiza mediante la reducción de la etiqueta al lema, se ignoran las *stop words*. Además consideran la posición de una palabra dentro de la propia etiqueta, en palabras compuestas, pero no especifican como la gestionan. La segunda versión de Taxomap está especialmente diseñada para ser más eficiente en grandes ontologías. Los autores acorde con un algoritmo trocean la ontología y estos bloques pasan por el anterior algoritmo, consecuentemente son unidos con sus correspondientes alineamientos. La evaluación se realizó en la OAEI de 2008.

### 3.6.16 MapPSO

MapPSO [11; 12] es un algoritmo que considera el problema del mapeado como un problema de optimización. El algoritmo está basado en el algoritmo de optimización de enjambre de partículas (*particle swarm optimization*, PSO) [27]. El algoritmo PSO usa una población de partículas para encontrar el parámetro óptimo respecto a unas funciones objetivo. Cada partícula representa una solución. En cada interacción del algoritmo, cada solución, cada alineamiento, evoluciona según los diferentes algoritmos de mapeado que integra el sistema. A nivel global, se mantiene un registro con todos los alineamientos. Este tipo de algoritmo es bastante versátil ya que las funciones objetivo pueden variarse o adaptarse a la naturaleza de la representación. El algoritmo trabaja incrementalmente lo que aporta dos tipos de beneficios: el algoritmo puede ser interrumpido en cualquier momento aportando soluciones y puede trabajar con unos resultados previos con el objetivo de ir refinándolos. Este tipo de algoritmos es paralelizable permitiendo una ejecución más eficiente.

El algoritmo utiliza diferentes medidas como funciones objetivo: nombres y etiquetas mediante la distancia de cadenas de caracteres [148], la distancia de etiquetas y nombres en WordNet -no indican que tipo de fórmula, ni que diferencia hay entre nombres y etiquetas-, la similitud por el espacio vectorial de los comentarios [135], la distancia jerárquica propagada en clases y en propiedades -no indican el cálculo-, y, finalmente, la similitud estructural por la firma de las propiedades -no indican el cálculo-. La combinación de estas medidas es mediante el operador OWA [66]. La evaluación se realizó en las campañas de OAEI 2008 y 2009.

### 3.6.17 RiMOM

RiMOM [93] es un algoritmo que implementa la selección dinámica de estrategias según la naturaleza de las ontologías implicadas. Está selección se basa en la variabilidad de dos tipos de medidas basadas en los nombres y en la similitud estructural las cuales ponderan los resultados de diferentes medidas básicas de similitud: léxicas y estructurales. Los autores opinan que no es siempre adecuado utilizar la misma configuración en la combinación de medidas y esta configuración ha de ser realizada de manera dinámica para flexibilizar el proceso.

Las medidas básicas son: la distancia de edición en la comparación de nombres de clases; una estrategia basada en espacio vectorial teniendo en cuenta las instancias y los comentarios de las clases; y a nivel estructural usan una variación del algoritmo de similitud *flooding* (SF) [106]. En este último algoritmo la ontología se transforma en un grafo etiquetado directo (*directed labeled graph*, DLG) según la conectividad de ambos grafos se establecen la semejanza de las nodos.

El tipo de alineamiento es de cuatro tipos: *exacto*, *estrecho*, *amplio* y parcialmente *superpuesto*. La evaluación se realizó en las campañas de OAEI 2006 y 2007; consiguió un primer y un tercero puesto, respectivamente.

### 3.6.18 ASMOV

ASMOV [64] es un algoritmo que usa similitudes lexicales y estructurales de dos ontologías para ir calculando iterativamente la similitud entre ellas y verificar que los alineamientos no contengan inconsistencias semánticas. Según los autores la naturaleza de las similitudes es similar a la propuesta realizada en OLA (3.6.4), donde la combinación de las diferentes medidas se basa en una suma ponderada.

Se ha definido similitud léxica, entre nombres y etiquetas, de la siguiente manera: la equivalencia, un 1, cuando dos nombres son idénticos; una similitud de 0.99 cuando un nombre es sinónimo del otro, con la peculiaridad de poder funcionar sin un tesauro externo WordNet o UMLS; una similitud de 0 cuando un nombre es antónimo del otro; y un valor de proximidad entre conceptos dentro de un tesauro acorde al trabajo de [94]. En caso de palabras compuestas, la similitud es computada por el número de palabras coincidentes. Los comentarios de las clases son procesados de manera independiente. Mediante una variación de la fórmula de Levenshtein (sección 3.3.1) se calcula el número de pasos necesarios para transformar una cadena a otra. Las similitudes léxicas se agrupan mediante una suma ponderada donde los pesos son fijados experimentalmente.

A nivel estructural se computa la relación jerárquica entre clases y propiedades mediante la similitud que guardan las clases o propiedades allegadas a padres e hijos en los diferentes niveles. Una vez determinadas los alineamientos estos se someten a una serie de condiciones lógicas con el objetivo

de encontrar alguna incongruencia. La evaluación se realizó en la campaña de OAEI 2008.

### 3.6.19 AgreementMaker

AgreementMaker [28][33] es un entorno con interfaz visual que da cabida a un repertorio de implementaciones de medidas básicas: léxicas y estructurales. Donde los diferentes métodos pueden combinarse o simplemente ejecutarse paralelamente. El resultado es una suma ponderada de los diferentes métodos que el usuario decida incluir. La ponderación de cada uno de ellos puede ser manual o automática acorde a sus experimentos.

Incorporan tres nuevas medidas: similitud base, similitud de herencia por descendientes y similitud por hermandad. Las tres medidas se definieron en un trabajo anterior [29].

La similitud base es una función por donde todos los conceptos son tratados bajo una serie de condiciones para establecer un valor de similitud. Al superar un umbral, este valor determina un alineamiento. Los criterios son léxicos por comparación de nombres, simples o compuestos, si coinciden en nombre se determina su equivalencia inmediata, si no es así se comprueba el número de palabras coincidentes en su definición mediante el uso de WordNet. La asignación de una acepción es inexplicada.

La similitud de herencia por descendientes se basa en una heurística que tiene en cuenta la propagación en diferentes niveles de jerarquía en WordNet la similitud de las clases padres. De la misma manera, la similitud por hermandad tiene en cuenta la similitud entre las clases hermanas, comparten la clase antecedente o padre. La evaluación se realizó en la campaña de OAEI 2007.

### 3.6.20 BLOOMS

BLOOMS [61] es un sistema para encontrar relaciones a nivel de esquema específicamente diseñado para operar en entornos distribuidos como es el caso de *Linked Open Data* (LOD) [34]. El sistema calcula alineamientos con la ayuda de datos disponibles en la red como es la Wikipedia. Esencialmente, encuentra alineamientos en jerarquías y necesita cualquier otra jerarquía donde poder ubicar estos conceptos. El funcionamiento de BLOOMS se basa en crear jerarquías llamadas *árboles*, y por consiguiente crean lo que denominan un *bosque* con todas las posibles combinaciones de elementos. Computan dentro del *bosque* la similitud de los diferentes *árboles* mediante procesos de simplificación y comparación de ramas, con la superación de un cierto umbral dan por establecida la correspondencia de equivalencia entre ambas clases. En 2009 participó en la campaña de OAEI.

[33] http://agreementmaker.org/
[34] http://linkeddata.org/

En la segunda versión, BLOOMS+ [62] se detallan una serie de aspectos o se incluyen unas medidas que en la anterior versión no estaban explicitas en la documentación. En este artículo se especifica la fórmula para detectar similitud entre los diferentes *árboles*, donde a medida que el árbol va siendo más grande la posible similitud va decreciendo exponencialmente. Incluyen una similitud contextual basada en la jerarquía de las clases implicadas. Si las superclases tienen o guardan aspectos parecidos entonces las clases pueden alinearse. La unión de ambas medidas se realiza mediante una suma ponderada. La evaluación de esta nueva versión consta de tres conjunto de datos: DBpedia, FreeBase y Geonames específicos de LOD junto con alineamientos obtenidos mediante el uso del sistema PROTON [30]

### 3.6.21 CODI

*Combinational Optimization for Data Integration* (CODI) [117] combina medidas léxicas y estructurales para detectar alineamientos entre clases, propiedades (objetos y tipos de datos), individuales. La combinación de medidas se lleva a cabo mediante la combinación de redes lógicas de Markov (MLN). Establecen relaciones lógicas, reglas de mapeado tanto en el T-box como en el A-box, basadas en clausulas de Horn. Entre las medidas léxicas mencionan el uso de Levenshtein, y a nivel estructural la equiparación de jerarquías.

### 3.6.22 Eff2Match

Los autores de la propuesta Eff2Match [23] centran sus objetivos en la eficiencia en términos del tiempo de respuesta. La partícula "Eff2" proviene de *Effective* y *Efficient* algoritmo de mapeado de ontologías. El algoritmo contrasta información de conceptos y propiedades para buscar la equivalencia semántica. Consiste en 4 pasos: la generación de anclajes de información, la generación de candidatos, la expansión de anclajes y un proceso iterativo de refinamiento. La generación de anclajes se basa en la búsqueda de *strings* exactamente iguales previamente preprocesados lingüísticamente. Usan una tabla de *hash* para optimizar la inserción y búsqueda de correspondencias entre *strings*. Las correspondencias producidas en la tabla pasan a formar parte de una selección de candidatos. En la selección se utiliza un modelo espacial de vector (*Vector Space Model*). Mediante la similitud de coseno entre los vectores de ancestros y de descendientes se comprueba la similitud de las anotaciones entre la jerarquía. Los resultados finalmente son normalizados a un valor [0,1]. La búsqueda de similitud no finaliza en estos dos pasos, se extiende en un tercero con la aplicación de nuevas técnicas en las correspondencias establecidas en la tabla de *hash*. En este proceso las palabras de los conceptos son troceadas en *tokens*. Cada par entre dos palabras es analizado en WordNet para determinar si son sinónimos. Este proceso se repite hasta que no quedan *tokens* en común entre ambas palabras, si no existen más *tokens* las palabras

son consideras equivalentes. Si quedan *tokens* entre ambas palabras, la similitud se determina por el mayor número de palabras. Por ejemplo, si la clase $C_1$ es *HeartEndocardium*, y $C_2$ es *Endocardium* y $C_2$ es un subconcepto de *Heart Part*, entonces ambos conceptos son equivalentes. Los autores indican que la palabra *Heart* de $C_1$ es "no informativa". La última etapa del proceso, consiste en determinar iterativamente la similitud de aquellas palabras que no están relacionadas en la tabla mediante un proceso de comparación de jerarquías.

## 3.7 Análisis

Se ha realizado una síntesis en función de unos criterios de todos los algoritmos anteriores para un estudio comparativo. Por razones obvias no puede abarcar en detalle cada uno de los algoritmos y ni siquiera las anteriores explicaciones podrían detallar cada una de sus peculiaridades. Esta síntesis está expuesta en las tablas 3.1 y 3.2. Los criterios están expuestos a continuación:

- Los nombres de las propuestas están en la primera columna. La nomenclatura (1) y (2) indica la versión del algoritmo. En algunas propuestas con varias versiones o modificaciones no se ha optado por realizar tal distinción a causa de la escasa diferencia en términos de nuestros criterios.
- Los *elementos* es la información que los autores pretenden manejar. Algunos trabajos sólo comparan individuales, otros las clases y el resto, el esquema en sí: clases, individuales, propiedades, axiomas, etc.
- Los *datos reales*, desde nuestro punto de vista, son la cuota superior de datos que necesitan para desempeñar su función. En este caso tenemos una serie de categorías incrementales:
  - taxonomía, las relaciones tratadas son jerarquías.
  - grafos, las relaciones forman redes más complejas que simples jerarquías.
  - ontología*, consideran las firmas de las de las propiedades.
  - ontología, consideran los diferentes tipos de funciones (ej. transitivas, reflexivas).
- *Medidas léxicas*. Se citan las medidas que han utilizado en el repertorio de medidas léxicas. Generalmente, emplean el nombre, las diferentes etiquetas, comentarios, que complementan a la clase y a su respectivo nombre.
- *Medidas estructurales*. Se citan las medidas que han utilizado . En el caso de emplear una heurística específica, se detalla con la palabra: "específica".
- *Medidas combinacionales*. El conjunto de medidas que utilizan para combinar las medidas léxicas, estructurales y/o el conjunto en sí sin importar la diferenciación.
- *Significado*. Todas aquellas propuestas donde el significado participa en la decisión de similitud se ha etiquetado con $y$. En aquellas propuestas donde se ha considerado el contexto, con $y*$.

- *Recursos externos* involucrados. Se incluye el conjunto de recursos involucrados, los principales son: WordNet, Wikipedia, DBpedia, y UMLS, aunque hay otros no comentados.
- *Evaluación*. Indican el repertorio de pruebas realizadas. Algunas se han detallado, otras son lo bastante específicas del problema para tal fin y, por último, se incluyen los años de las campañas de OAEI donde participaron.
- *Representación*. La forma de representar el alineamiento, es habitual considerar la equivalencia o sinonimia con un valor entre 0 y 1. En otros casos, es mediante algún tipo de información extra sobre la relación: subsunción, contradicción, etc.
- *"Se centran en"*. Representa la temática de algunas propuestas ya que centran la mayoría del contenido a explicar ese aspecto.

| System | Year | Elements | Real Data | Lexical | Structural | M. Combination | Meaning |
|---|---|---|---|---|---|---|---|
| $COMA^{(1)}$ | 2002 | schema | taxonomy | qGram, EditDistance, SoundEx | specific: children, leaves | Average, Dice coefficient | - |
| $GLUE$ | 2003 | instances | taxonomy | | Jaccard | Weighted Sum | - |
| $S-Match^{(1)}$ | 2004 | schema | ontology | qGram, EditDistance | Hierarchy distance | SAT | y |
| $OLA$ | 2004 | classes | graph | | specific | Weighted Sum | - |
| $Falcon-AO^{(1)}$ | 2005 | schema | graph | edit distance, TF/IDF on VSM | same properties | Cosine Similarity | - |
| $COMA++^{(2)}$ | 2005 | schema | ontology* | qGram, EditDistance, SoundEx | specific: children, leaves | Average, Dice coefficient | - |
| $MoA$ | 2005 | schema | taxonomy | | specific rules | | y* |
| $SAMBO$ | 2006 | schema | ontology | qGram, EditDistance | Hierarchy distance | Weighted Sum | - |
| $Gerome/Suite$ | 2007 | schema | graph | | Similarity Flooding / SP. Rules | | - |
| $AROMA$ | 2007 | schema | taxonomy | JaroWinkler | association rules | | - |
| $LILY$ | 2007 | classes | graph | | subgraphs / specific | specific | - |
| $SEMA$ | 2007 | schema | ontology | cluster, Kullback-Leibler divergence, VSM | specific rules on hierarchical, properties | | - |
| $DSSim$ | 2007 | classes | ontology* | | Jaccard* | Dempster's rule | - |
| $Prior/+$ | 2007 | classes | taxonomy | Levenshtein, VSM, TF/IDF on doc | Structural propagation of Lex | NN-Based Constraint Satisfaction | - |
| $SeSa$ | 2008 | schema | graph | | specific rules matching graph | | - |
| $Falcon-AO^{(2)}$ | 2008 | schema | ontology | VSM on profile information | specific rules matching graph | Weighted Sum and dynamic WS | y* |
| $TaxoMap$ | 2008 | classes | taxonomy | Label %, | Hierarchy % | | - |
| $MapPSO^{(1)}$ | 2008 | schema | ontology | SMOA string distance, WN distance | Hierarchy distance, signature properties, ? | Weighted Sum | - |
| $RiMOM$ | 2009 | schema | ontology* | Edit distance, VSM | Similarity Flooding* | Weighted Sum with dynamic weights | - |
| $ASMOV$ | 2009 | schema | ontology | Rule specific | OLA*, specific rules | Weighted Sum | - |
| $AgreementMaker$ | 2009 | schema | ontology | dynamic | dynamic | Weighted Sum | - |
| $CODI$ | 2009 | schema | ontology | Levensthein | Hierarchy distance ? | Markov networks | - |
| $S-Match^{(2)}$ | 2010 | schema | ontology | qGram, EditDistance | Hierarchy distance | SAT | y |
| $BLOOMS/+$ | 2010 | classes | taxonomy | | specific | Weighted Sum | y* |
| $MapPSO^{(2)}$ | 2010 | schema | ontology | SMOA string distance [3], WN distance | Hierarchy distance, signature properties, ? | Weighted Sum | - |
| $Eff2Match$ | 2010 | schema | ontology | String equivalence | SVM, hierarchical weighted | Exact process, Weighted Sum | - |

Cuadro 3.1: Posición de los sistemas analizados acorde a los criterios

| System | Meaning | Resources | Evaluation | Representation | Focus on |
|---|---|---|---|---|---|
| $COMA^{(1)}$ | | | Specific | 0..1 | |
| $GLUE$ | | | University Courses | 0..1 | Machine Learning |
| $S-Match^{(1)}$ | y | WordNet | University Courses, Directory | Relationships | |
| $OLA$ | | | | 0..1 | |
| $Falcon-AO^{(1)}$ | | | OAEI 2005 | 0..1 | |
| $COMA++^{(2)}$ | | | Specific | 0..1 | |
| $MoA$ | y* | | | Relationships | |
| $SAMBO$ | | WordNet; UMLS | Specific | 0..1 | Biological Ontologies |
| $Gerome/Suite$ | | | | 0..1 | Metamodels |
| $AROMA$ | | | University Courses, OAEI 2005 | Relationships | |
| $LILY$ | | | OAEI 2007 | 0..1 | |
| $SEMA$ | | | OAEI 2007 | 0..1 | |
| $DSSim$ | | | OAEI 2006 and 2007 | 0..1 | |
| $Prior/+$ | | | OAEI 2006 and 2008 | 0..1 | |
| $SeSa$ | | | Specific | 0..1 | |
| $Falcon-AO^{(2)}$ | y* | | OAEI 2006 and 2007 | 0..1 | |
| $TaxoMap$ | | | OAEI 2008 | Relationships | |
| $MapPSO^{(1)}$ | | WordNet | OAEI 2008 | 0..1 | Particle swarm optimization |
| $RiMOM$ | | | OAEI 2006 and 2007 | Relationships | Dynamic strategic |
| $ASMOV$ | | WordNet | OAEI 2007 and 2008, specific | Relationships | Semantic Verification |
| $AgreementMaker$ | | | OAEI 2007, specific | 0..1 | |
| $CODI$ | | | OAEI 2009 and 2010 | 0..1 | |
| $S-Match^{(2)}$ | y | WordNet | University Courses, Directory | Relationships | SAT |
| $BLOOMS/+$ | y* | Wikipedia, Dbpedia, | OAEI 2009 | 0..1 | |
| $MapPSO^{(2)}$ | | WordNet | OAEI 2009 | 0..1 | Cloud Computing |
| $Eff2Match$ | | WordNet | OAEI 2010 | 0..1 | |

Cuadro 3.2: Posición de los sistemas analizados acorde a los criterios (*continuación*)

De los siguientes datos concluimos una serie de puntos: (i) Hasta aproximadamente el 2007, las propuestas no se centran en ontologías, es decir, no consideran OWL como el lenguaje específico sobre el cual mapear. Una de las posibles causa sea que OWL no se convierte en estándar de W3C[35] hasta 2004, comenzando su influencia en diferentes áreas investigación. (ii) A partir del año 2005 se aprecia la influencia de OAEI en la evaluación y se consolida definitivamente en 2007. (iii) De las 22 propuestas[36], 6 representan sus resultados mediante un valor numérico y el tipo de relación, al menos, consideran la equivalencia y subsunción. Por el número de las propuestas, no se puede asegurar pero tal vez esté causado por la influencia de OAEI en el requerimiento de aceptar los alineamientos mediante el formato definido en *Alignment API*. (iv) WordNet es el recurso externo más utilizado. (v) En la combinación de las diferentes medidas, la estrategia más utilizada es la suma ponderada. (vi) Observando la columna de *datos reales* podemos deducir los siguientes hechos. De un total de 22 propuestas[36], 11 (taxonomías y grafos) podían resolverse sin la necesidad de estar representadas mediante ontologías y 11 de ellas están diseñadas específicamente para ontologías; 3 propuestas (ontologías*) no requieren de tipos especiales de relaciones y 10 utilizan el repertorio de constructores de OWL. Ninguna tiene en cuenta los cuantificadores existenciales y universales. Propuestas como Taxomap [19] directamente predisponen que la naturaleza actual de los modelos semánticos son meras estructuras jerárquicas sin definiciones de propiedades e instancias. Desde nuestro punto de vista, se puede achacar al uso incorrecto de los autores de estas "ontologías" de la palabra ontología en vez de utilizar el término correcto: taxonomía. (vii) De las 22 propuestas[36] presentadas, 4 han considerado el contexto y el significado. Sólo en una propuesta, S-Match, el significado ha participado en el cálculo.

Un algoritmo de mapeado de ontologías, ha de ser un algoritmo de mapeado semántico por ser coherente con el tipo de representación tratada y ha de contemplar necesariamente medidas de *similitud semántica*, es decir, no debería de considerar medidas de similitud léxicas o "sintácticas"[37]. Obviamente, en devenir por la calidad de los resultados a un mayor número de criterios mayor será, en teoría, la calidad de los mismos. No hay que olvidar que la semántica se sustenta en la sintaxis y ésta a su vez, en el léxico.

Desde las afirmaciones planteadas en la sección 2.1, la similitud semántica implica similitud léxica pero la implicación no es recíproca, es decir, la similitud léxica no implica similitud semántica. Una propuesta de algoritmo con una única medida de *similitud semántica* es formalmente más adecuado que muchas de las propuestas aquí citadas.

[35] http://www.w3.org/2004/OWL/

[36] Hemos descartado las versiones

[37] Para mantener una terminología lingüística la simplificación de la sintaxis en lenguajes computacionales está proporcionada por los constructores que definen la estructura. En este caso, la similitud sintáctica se considera similitud estructural.

Una ontología es una representación semántica de un dominio, utilizada para representar esquemas, entonces hay que aclarar el tipo de medidas que son semánticas para tener un algoritmo de mapeado semántico. En esencia, trabajar con información semántica implica trabajar con el significado de las palabras, frases, etc. [88]. En el estudio presentado por [139], llevan a cabo un estudio respecto a la similitud o como denominan la proximidad semántica, la cual lo definen como: "un intento de caracterizar el grado de similitud semántica entre dos objetos usando la semántica real del mundo (RWS) del objeto. Siendo una medida cualitativa de la medidas introducida por [138]: equivalencia semántica, relación semántica, relevancia semántica y parecido semántica." De esta manera llegan a la definición de proximidad semántica, mediante una tupla con la siguiente información:

$$semPro(O_1, O_2) =< Context, Abstraction, (D_1, D_2), (S_1, S_2) > \qquad (3.24)$$

donde $D_i$ es el dominio de $O_i$ y $S_i$ es el estado de $O_i$. "El contexto de un objeto es el vehículo primario para capturar la semántica real del mundo del objeto."

También, en la integración de base de datos, [13] introduce un mapeado de esquemas contextual, donde a partir del mapeado del esquema se infieren ciertas vistas sobre las tablas. Introducen nuevas técnicas a la propuestas del trabajo de [109] basadas en la aproximación de "Strawman" donde una serie de categorías determinan el grado de mejora de cada uno de los alineamientos.

Sin entrar en la caracterización de su contexto y de lo explicado anteriormente en el capítulo anterior, el significado de una clase, propiedad o cualquier otro elemento constitutivo de una ontología no está expresado por el tipo o función "sintáctica", sino por el *todo* o nombrado con menos ambigüedad como contexto.

La mayoría de las propuestas consideran que dos elementos al tener el mismo nombre, independiente de haber pasado por un proceso pre-lingüístico, comparten una relación de equivalencia o sinonimia. En la suma ponderada, o en la heurística de combinación, el peso asignado a esta medida es superior al resto de pesos. La denotación de la palabra *acepción*[38] determina la existencia de más de un significado por concepto. Cuando se establece una equivalencia por similitud léxica existe la incertidumbre del error, de la no correcta correspondencia de acepciones. Se puede alegar que los conceptos representados en dos dominios específicos no presentan variabilidad de acepción. Es decir, los conceptos por su nombre rara vez son ambiguos. También se puede alegar que dos dominios idénticos presentan conceptos con las mismas acepciones por lo que las comparaciones léxicas y/o sintácticas son suficientes. Estas dos alegaciones ocurren en el conjunto de pruebas de OAEI. Mismas comparaciones entre dominios afines y específicos: conferencias, publicaciones y directorios. No existe la necesidad de gestionar el significado para obtener unos resultados

[38] "Cada uno de los significados de una palabra según los contextos en que aparece." fuente RAE

aceptables en el *benchmark* más utilizado. Creemos que es necesario incluir el significado en los procesos de mapeado para poder discernir entre clases iguales a nivel léxico de diferentes dominios. De esta manera, la aplicabilidad del algoritmo no está sometida a dominios afines que no son representativos de la naturaleza de la web.

Por tal motivo es necesario contemplar técnicas de desambiguación, específicas del descubrimiento de acepciones en palabras (*word sense desambiguation*) como tarea y métrica básica del mapeado de ontologías y del mapeado semántico. En la figura 3.6 vemos una serie de fases básicas como el tratamiento pre-lingüístico y la gestión de recursos externos para la fase fundamental de interpretación del significado.

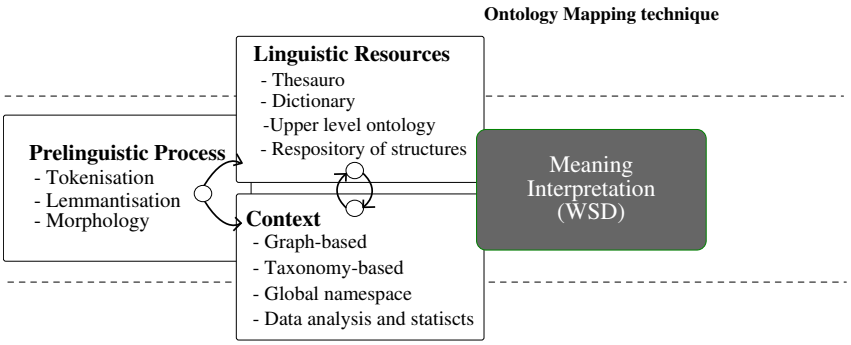


Figura 3.6: Técnica de desambiguación de elementos como punto central de un OM algoritmo

Podemos plantear dos tipos de posturas, aplicables a las propuestas marcadas con $y*$ e $y$ respectivamente. La primera considera la acepción como algo abstracto e inocua al algoritmo ya que dos clases implicadas representarán lo mismo si sus contextos son iguales. La segunda postura, es operar con la acepción explícitamente aplicando cualquier medida aplicable a ella y descubrir la correspondencia entre ambas. El segundo caso requiere del primero, el descubrimiento de la acepción requiere del contexto y la representación de la misma algún tipo de formalismo o recurso como es un diccionario, tesauro, etc. Por el contrario, la primera puede o no usar recursos externos para complementar la información contextual.

Análogamente a la primera postura, el contexto sin importar el significado, puede ser utilizado como una justificación para el resto de propuestas ya que tuvieron en cuenta información *contextual*. Al no poder demostrar esta predisposición a medir el contexto, las podemos etiquetar o degradar a propuestas de similitud sintáctica, lo cual no quita relevancia a su trabajo y aportación científica.

Ambas propuestas son validas y pueden considerarse como propuestas semánticas. Por tanto, una medida semántica es aquella que contempla el contexto y/o significado de los elementos a mapear.

De las dos posturas planteadas con o sin significado se dispone de más información para las posteriores medidas si se utiliza el significado y además podemos reducir la complejidad computacional. Consideremos la primera propuesta, manejar el contexto. Averiguar el contexto del resto de clases relacionadas mediante posibles soluciones como la recursividad o gestionando el posible repertorio de combinaciones lo convierte en un problema de complejidad NP. Por el contrario, al manejar el significado de la clase existe la posibilidad de encontrar clases no ambiguas independientemente del contexto lo que permite simplificar y finalizar el proceso anterior propagando esta información. Desafortunadamente, la incertidumbre está presente en ambas.

Otra cuestión a destacar es la perdida de información del proceso a equiparar dos conceptos mediante un valor numérico. Este hecho es innecesario cuando sólo se considera la relación de equivalencia, o de sinonimia o aquella que cada propuesta haya considerado. La no definición transforma el resultado en ambiguo para futuros procesos de reutilización. Como se ha analizado anteriormente, la causa más probable es la necesidad de proveer los resultados del *benchmark* en el estándar definido en *Alignment API* por y para las campañas de OAEI . Otras propuestas informan el tipo de relación que guarda ese grado de similitud potenciando su reutilización y adaptación en diferentes niveles de la aplicación. Es necesario promover un estándar de representación semántico de alineamientos.

## 3.8 Sumario

Si hacemos una similitud de esta área con la lingüística, vemos los grandes solapamientos naturales que hay entre ambas. La similitud semántica requiere de similitudes léxicas y estructurales, de la misma manera que el nivel semántico de una lengua se sustenta en los niveles léxicos y sintácticos. La gran diferencia estriba en la innecesaria capacidad de entender el contexto. La interpretación o el modelo lógico subyacente en los modelos semánticos se define mediante el lenguaje, donde recae la semántica real de la representación. Aun así, cualquier lenguaje no elimina la necesidad de conocer el significado real de los elementos cuando se comparan modelos de diferentes dominios.

Las propuestas actuales combinan medidas léxicas y sintácticas con un conjunto de heurísticas amplias y flexibles que dificultan su caracterización. Nuestra descripción se basa en una serie de categorías siguiendo la línea fijada por [126] y que posteriormente siguió y amplio [41]. De entre todas las propuestas analizadas son pocas las que consideran el contexto o incluso el significado como parte esencial del proceso.

Es necesario contemplar el contexto y el significado real de los elementos de una ontología como etapa esencial en el proceso de mapeado. Si relaciona-

mos este requerimiento con otras áreas computacionales, estamos dentro del conjunto de técnicas denominadas: desambiguación del sentido de las palabra (Word Sense Disambuation, WSD), donde ahora el corpus es una representación ontológica.

Concluimos con el esquema final de nuestra propuesta en la figura 3.7. Manifestamos la necesidad de identificar y usar el contexto o el significado de los elementos representados para la realización efectiva de un proceso de mapeado. En este trabajo, muchos procesos son comunes y han sido aplicados en diferentes propuestas: tareas de preprocesado lingüístico, organización de datos, medidas léxicas, estructurales y combinacionales. En ninguna de ellas se usa el significado como eje central donde desarrollar un algoritmo no iterativo de reglas para encontrar de manera efectiva alineamientos. En función del significado expandimos la información disponible sobre el cruce de palabras con los constructores utilizados y los recursos externos utilizados.

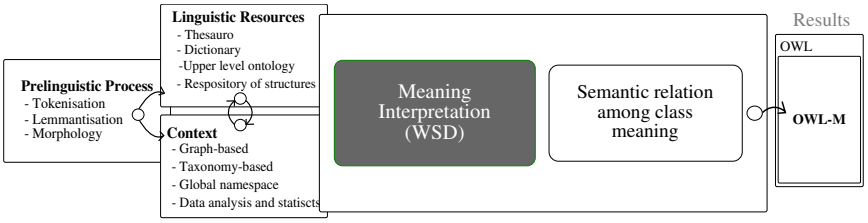


Figura 3.7: Estructura de OMoCC

# Parte II

# Contribution

# 4

# Los fundamentos del algoritmo OMoCC

Nuestro algoritmo está basado en una serie de hipótesis donde el contexto de las clases desempeña un papel central sobre los resultados. El acrónimo *OMoCC* proviene de su definición en inglés: *Ontology Mapping based On Classes Context*.

Los puntos claves se han englobado en dos ideas, donde se explica la problemática que da origen a las hipótesis. La primera idea está relacionada con el significado de los elementos de la representación dentro del contexto, tal como se analiza en el anterior capítulo. La segunda se centra en la manera de representar alineamientos. Ambos puntos son los pilares del funcionamiento del algoritmo OMoCC.

## 4.1 El significado de los elementos

En el análisis del trabajo relacionado es necesario considerar el significado y el contexto durante la equiparación de elementos. Tal como se pudo constatar con los trabajos de [75; 109; 127; 128]. De esta manera, la medida aplicada es una medida semántica que puede o no ofrecer mejores resultados dependiendo del *benchmark* o de la calidad de la ontología -en términos descriptivos del uso correcto y completo de los constructores-. Comparar dominios diferentes pero con ciertos solapamientos mediante medidas estructurales y léxicas, no proporciona resultados precisos. En este sentido, defendemos que uno de los primeros pasos en la equiparación de elementos entre ontologías sea la interpretación y el uso de medidas relacionadas con el significado y el contexto.

Conocer el significado de una palabra, es decir, encontrar la acepción adecuada dentro de la representación, es un problema ampliamente estudiado en el tratamiento de lenguajes naturales llamado *Word Meaning Disambiguation* (WSD) y considerado una tarea fundamental en la traducción automática (*Machine Translation*) a finales de los 40 [99; 114]. WSD es considerado como un problema completo en Inteligencia Artificial a causa de la gran variedad de factores implicados y a su alta interdependencia. Destacamos entre estos

factores el volumen de palabras implicadas, la representación a diferentes niveles (léxico, sintáctico y semántico), la escasez de medios lingüísticos donde comparar el posible repertorio de acepciones, la limitación de los constructores y la claridad de la representación, entre otros. Típicamente, los algoritmos de WSD trabajan sobre corpus de estructura textual: parágrafos, líneas, documentos, etc. En nuestro caso, las fuentes son esquemas ontológicos donde la tarea de desambiguación recae sobre las clases como elementos principales de la representación. Esta disciplina permite asentar conocimientos sobre otros lenguajes al aquí utilizado.

Tal como vimos en la sección 2.3.1, el lenguaje implicado y sus características como la expresividad, el modelo de interpretación y la estructuración de los constructores, van a determinar el posible repertorio de medidas para discernir la acepción y, por ende, esta decisión afectará a la correcta adecuación de alineamientos

Veamos tal afirmación con una serie de ejemplos, basados en tres tipos de representaciones. La primera representación son las nubes de conceptos (*tag cloud*), donde disponemos de dos posibles medidas: la frecuencia de aparición y el nombre. Si utilizamos la frecuencia ($TF/IDF$) como medida de coincidencia entre dos palabras los resultados no serían los esperados. Tenemos que recurrir a medidas de similitud basadas en la etiqueta del nombre o combinar ambas para tener, quizás, mejores resultados. Podríamos considerar como contexto el resto de palabras de la nube pero son datos dispersos al no disponer de relaciones entre las etiquetas.

Una segunda propuesta son los modelos entidad-relación. De ellos podríamos obtener la siguiente información: conjunto de tablas, listado de atributos con forma de tupla (nombre y tipo), información de control como son los identificadores y a partir de ellos se puede obtener las posibles relaciones entre tablas. A la hora de equiparar información tendremos el mismo problema anterior si sólo confiamos en la estructura relacional entre tablas por compartición de identificadores. Necesitamos recurrir a los nombres de cada tabla pero esta vez los atributos y las relaciones entre tablas pueden ayudarnos a discernir peculiaridades. La existencia de estas relaciones incrementa el posible contexto de cada tabla dentro del dominio de la representación. Definitivamente, un modelo entidad relación provee mayor volumen de información que un modelo basado en nubes de conceptos.

La última propuesta son los modelos basados en RDF-Schema. En ellos disponemos de información básica (clases, propiedades de objetos, propiedades de tipo de datos, individuales, dominio, rango, subclases y subpropiedades) y, gracias al modelo lógico subyacente, de una serie de hechos implícitos. Básicamente, son nuevos hechos basados en subsumir un elemento a una clase o a una propiedad a partir de una relación de jerarquía o, por definición de una propiedad mediante el dominio y el rango. OWL extiende el repertorio de constructores de RDF y el modelo lógico al mismo tiempo. Los modelos semánticos comparados con el resto de representaciones aportan más medidas para tales fines. Entre las medidas útiles tendremos información estructural,

relacional u otras relacionadas con los individuales, axiomas, restricciones, etc. Sin embargo, estas medidas no darán con soluciones exactas sino recurrimos de nuevo al nombre de las clases y de las relaciones. Para ilustrar tal afirmación pongamos un caso en OWL donde equiparamos una clase por su información estructural. En un esquema existe la definición de una clase $A_1$ como la restricción $A_1 \subseteq (B_1 \sqcup \exists R_{prop.}.D_1)$. En otro esquema existe la misma estructura axiomática. El resultado no puede ser la equiparación de ambos sin tener en cuenta el significado de las etiquetas, del resto de los elementos que participan, el contexto, y la participación de otras medidas.

En resumen, el lenguaje de representación determina la posible información útil con la que cotejar acepciones y establecer alineamientos. Podemos diferenciar parte de esta información como la información contextual de cada elemento, es decir, las relaciones con el resto de elementos de la representación. Con la incorporación del proceso de desambiguación en el proceso de mapeado, el descubrimiento de acepciones de los elementos permite comparar dominios diferentes con lo que aumenta la precisión de los alineamientos sin apenas incrementar el uso de recursos computacionales por las sinergias de ambas tareas y como se puede ver en las propuestas que usan recursos externos pero no consideran el contexto.

## 4.2 La representación de los resultados

Otra problemática presente en un proceso de mapeado es la integración de los resultados. Resolver que dos elementos son iguales, con o sin un grado de similitud, y no introducir este hecho en la unión de ambos modelos hace que todo el esfuerzo sea perecedero. Si se opta por integrar los resultados en la fusión de ambos modelos también ha de gestionarse la inclusión del hecho con el contexto de cada elemento. Pongamos un ejemplo, un proceso de mapeado decide que los elementos $A_1$ y $A_2$ son equivalentes, sin embargo, $A_1$ se define como $\subseteq (B_1 \sqcup \exists R_{prop.}.D_1)$ y en otro modelo, $A_2$ se define como $\subseteq B_2$. La fusión de ambos modelos creará un estado de incongruencia entre ambas restricciones. Tal vez, $B_1$ y $B_2$ sean equivalentes y simplifique el hecho de tratar sólo con la restricción sobre la propiedad pero sigue existiendo un problema de integración. Esta representación de la integración deriva en otra serie de problemas relacionados con la utilización de los resultados por parte de las mismas herramientas que generan o manejan los modelos. Si los resultados no son representados en un lenguaje compatible, las capas superiores de la aplicación tendrán que usar herramientas específicas para tratarlos, con el consecuente gasto de recursos. Por eso hemos optado por representar el mapeado de modelos y la integración de alineamientos con el mismo lenguaje.

Tanto la desambiguación de elementos como la representación de alineamientos son dos grandes tareas tratados en esta tesis. Durante la desambiguación trataremos otro aspecto tan importante como es la gestión de palabras

compuestas. Para facilitar la lectura del documento, el resto de ideas se introducen y se desarrollan a lo largo de los siguientes capítulos. El desarrollo del algoritmo es, en si, lineal al desarrollo de cada uno de los puntos.

# 5

# El significado: la acepción de cada clase

En este capítulo vamos a explicar nuestra metodología para utilizar la información contextual, la que nos proporciona la acepción, para obtener una medida, realmente, semántica.

## 5.1 Descubrimiento del significado

La idea de esta parte del algoritmo es determinar la acepción de una clase y representarla en la ontología. La acepción del resto de elementos no se representa pero sí es determinada e influye en la elección de la acepción de la clase. Entonces veremos como determinar la acepción en función de la representación, a partir de la ontología, y el resto de fuentes externas que hemos considerado. Una vez identificada la acepción, ésta será la medida básica del algoritmo de mapeado y guiará el establecimiento de alineamientos: "dos conceptos son iguales si tienen la misma acepción independientemente de sus relaciones, nombre o comentarios". Esta heurística [91] aplicada a OMoCC se define en el capítulo 7.

El repertorio de acepciones de una palabra se obtiene mediante WordNet (ver sección 3.3.4). WordNet se ha empleado como diccionario para identificar palabras correctamente escritas y como tesauro por su estructura y por el conjunto de *synsets*. Gracias a su estructura podemos plantear medidas de similitud respecto a una ontología que con el resto de tesauros -Roget y Moby- no podrían realizarse.

El proceso de descubrimiento del significado solapa alguna de las tareas del proceso de mapeado por lo que al mismo tiempo podemos aprovechar las sinergias comunes entre ambos. Procesos comunes en ambos son el análisis de elementos, el análisis lingüístico, léxico y estructural. Además, la estructura de datos creada para gestionar las acepciones es aprovechada en los siguientes pasos propios del mapeado.

## 5.2 Nomenclatura

Con el objetivo de facilitar la exposición de hechos, se ha optado por utilizar una nomenclatura de la estructura lingüística basada en la información disponible en diccionarios, tesauros y ontologías. Especialmente, está basada en la estructura de WordNet. Esta nomenclatura permite representar las relaciones entre palabras entre las diferentes fuentes, permitiendo la construcción de un camino entre palabras en las mismas y entre las diferentes fuentes. Un camino no es más que un conjunto de propiedades entre diferentes palabras. Este camino permite representar, calcular y valorar la idoneidad del mismo como posible alineamiento.

Una clase de una ontología la definimos como $C$. El nombre de la clase lo denominaremos concepto. El concepto está formado por una o más palabras, las cuales las llamaremos *palabras simples*, siendo $C \equiv sW_i$. Más de una palabra simple constituye una palabra compuesta, posteriormente veremos como se analiza este tipo de formaciones. Las palabras simples han sido procesadas lingüísticamente tanto para su detección como para su posterior manejo. Cada palabra simple tiene una o varias acepciones $C \equiv sW_i : S_i$, donde: $S$ tiene su origen de la palabra *sense* usada en WordNet, ':' significa "tener" y el $i$-índice significa "elemento del conjunto", sin relación entre los diferentes índices.

Cada acepción dentro de WordNet tiene una definición llamada *gloss*, esta definición escrita en lenguaje natural esta compuesta por una serie de palabras denominadas términos $sW_i : S_i : t_i$, siendo lo mismo que su traspuesta $t_i : S_i : sW_i$. Cada acepción tiene un conjunto de palabras relacionadas semánticamente por WordNet y por la ontología $S_i : Hpon_i, Hper_i, Syn_i, Ant_i, Mer_i, Hol_i$. Además cada palabra de la relación semántica $(Hpon_i, Hper_i, ,...)$ es en si otra palabra pudiendo analizarse como una palabra simple, por ejemplo $Hpon_i \equiv sW_j$.

En el código 5.1 presentamos parte de un ejemplo relacionado con la palabra `arts`. La acepción proviene de WordNet[1] (figura 3.3). La clase `arts` se descompone en una acepción, preprocesada lingüísticamente -en este caso es idéntica-. La primera acepción ($sW_1$) contiene dos sinónimos ($Syn_1$ y $Syn_2$), un *gloss* ($t_{1.,9}$), una serie de superclases: *artificial flowers* y *commercial art*. Finalmente, otros dos conceptos son añadidos: *artistic* y *creation*. Por motivos, de espacio no se desarrolla todo el conjunto de relaciones posibles.

Código 5.1: **Nomenclatura aplicada a la palabra *arts***

$C = \{\texttt{arts}\}$
$C : sW_1 = \{\texttt{art}\}$
$sW_1 : S_1 : Syn_1 = \{\texttt{fine art}\}$
$sW_1 : S_1 : t_{1.,9} = \{\texttt{the products of human creativity, works of art collectively}\}$
$sW_1 : S_1 : Hpon_1 = \{\texttt{artificial flower}\}$
$sW_1 : S_1 : Hpon_2 = \{\texttt{commercial art}\}$
$sW_1 : S_2 : Syn_1 = \{\texttt{artistic creation}\}$
$C_j : sW_1 = \{\texttt{artistic}\}$
$C_j : sW_2 = \{\texttt{creation}\}$

[1] http://wordnetweb.princeton.edu/perl/webwn?s=art

## 5.3 Metodología

Para descubrir el significado de las clases analizamos el significado de los conceptos, de las propiedades entre clases (*objectProperties*) e individuales. Exceptuamos, las propiedades de datos *datatype properties*, las restricciones, la cardinalidad y el tipo de propiedad (funcional, transitiva, etc.). Al finalizar el proceso sólo se deja constancia escrita del significado de las clases mediante anotaciones sobre el identificador único de cada acepción utilizado en WordNet. Los individuales al representar entidades específicas y/o nombres propios rara vez tienen un significado generalista por lo que no existe la necesidad de guardar tal acepción.

[6] proponen también la necesidad de anotar el significado de una clase, para su posterior uso en algoritmos de mapeado. Definen una *distancia* entre un par de clases como el número de ejes/pasos/aristas que forman el camino más corto entre ellos. Estas clases próximas a la principal son llamadas vecindario y entran en un radio $r$. Calculan la probabilidad de una acepción, $C_i$, sea correcta a un concepto $C$ mediante la afinidad, $a$, a otro concepto vecino $D$: $a(C_i, D) \in [0, 1]$. De todo el conjunto de acepciones la que obtenga mayor afinidad con el conjunto de clases vecinas es la supuestamente más correcta. La afinidad depende también de la distancia y el radio entre la vecindad, lo cual afectará al peso de la media ponderada de la acepción. La fórmula es la siguiente:

$$a_{total}(C_i) = \frac{w_1 \sum_{j=1}^{|R_1|} a(C_i, D_{1,j}) + w_2 \sum_{j=1}^{|R_2|} a(C_i, D_{2,j}) + \ldots + w_n \sum_{j=1}^{|R_n|} a(C_i, D_{n,j})}{w_1 \mid R_1 \mid + w_2 \mid R_2 \mid + \ldots + w_n \mid R_n \mid} \tag{5.1}$$

donde $n$ es el radio de vecindad, $\mid R_x \mid$ el total de vecinos a esa distancia y $w_x$ son los pesos correspondientes a cada distancia. Fijan el radio experimentalmente a una distancia no mayor a 5. El coeficiente de afinidad lo han basado en el trabajo presentado por [165]. La taxonomía de WordNet es interpretada como un grafo donde los vértices son los *synsets* y cada relación es un eje. A cada eje se le asigna un peso entre 0 y 1 acorde al tipo de relación. Se consideran todos los posibles caminos entre el vértice objetivo y todos los correspondientes vértices para todas las acepciones. Los pesos son multiplicados a lo largo del camino y el mayor de todos ellos representa la afinidad entre ambos. Este trabajo presenta una similitud al trabajo de Resnik pero con la diferencia de una ponderación del camino (ver 3.3.5). En su estudio han aplicado otra técnica de afinidad basada en el trabajo de [95]. La selección de la acepción se basa en el solapamiento de términos con los sinónimos, las palabras del *gloss*, los sinónimos de todos los hipónimos y todas las palabras de los hipónimos, consiguiendo '4x4' pares de comparaciones. Para la evaluación se utilizaron tres ontologías, aunque la comparación con un estándar -un conjunto de acepciones definido- no queda explícito en la documentación.

Las ideas de Banek *et al.* coinciden con las nuestras en la necesidad de calcular la acepción. Sin embargo su finalidad es la automatización de las anotaciones del resto de las aproximaciones donde la anotación de la acepción es realizada por *personas*. Citan tres propuestas como S-Match [48] (ver 3.6.3), H-Match [21] y OntoGenie [122], todas ellas de 2003 y 2004 como supuestos casos de anotaciones manuales.

### 5.3.1 Preprocesado lingüístico

El primer paso de nuestra metodología es el preprocesado lingüístico. En el caso de ontologías, a diferencia de su aplicación en una estructura tradicional (explicado en pág. 25), se ha aplicado los siguientes procesos: *tokenization*, capitalización y un analizador lingüístico (*stemming*). Estos procesos se aplican para cualquier elemento que no tenga significado en el tesauro. Por consiguiente, palabras compuestas que aparezcan en el tesauro se tratan como palabras simples, por ejemplo `hot-dog`.

- ***Tokenization***. Consiste en la división de nombres en unidades llamadas *tokens*. Cada elemento separador depende del diseñador por lo que puede ser un conjunto de letras, de números, de símbolos de puntuación u otra marca de separación como el uso de la capitalización de la primera letra de cada palabra. Esta ultima nomenclatura se denomina *CamelCase* y es la que hemos considerado aunque también tratamos otro tipo de separadores. Por ejemplo, los *tokens* del concepto `pizza_toppings` son: `pizza`, '_' , y `toppings`; de la palabra `year1998` son: `year` y `1998`. No es necesario utilizar entornos de programación para resolver esta funcionalidad ya que detectar este tipo patrones es fácil mediante el establecimiento de patrones. Ignoramos los *tokens* con números por que suelen ser usados para identificar inequívocamente individuales comunes. Al realizar esta suposición, consideramos la acepción de dos clases iguales si presentan el mismo nombre y diferentes números, como por ejemplo `wine100` y `wine101`.
- **Capitalización**. Después del proceso de *tokenization* las palabras simples pasan a través de un proceso de capitalización. La cuestión radica en si dos palabras que comparten identificadores idénticos pero con ciertas letras capitalizadas se han de considerar iguales. Nuestro suposición es afirmativa, es decir, son iguales. Con esta suposición dos tipos de *tokens* como `Curry`, de `Tim Curry`, y `curry`, de especia, comparten el mismo significado. Este tipo de casos se da en nombres propios. Estos supuestos se evitan recurriendo a una base de datos externa con documentación relacionada con personas, lugares, inventos, metodologías, etc. Gracias a ello la palabra tendría sentido antes de pasar por el procesado lingüístico. En este algoritmo no se ha considerado una base de datos externa específica para detectar estos casos.
- **Analizador de Porter** (ver sección 3.2.1), hemos utilizado el analizador de Porter para reducir las definiciones (el *gloss*) de los conceptos aparecidas en WordNet. El resto de nombres de conceptos, clases e individuales

es preferible mantener la nomenclatura original para evitar equiparar conceptos como por ejemplo, `operating` y `operative`, que son reducidas al mismo término `operat`.

El preprocesado se aplica a cualquier elemento de la ontología siempre y cuando no tenga significado por si mismo. Además, se aplica a las definiciones de los conceptos dentro de WordNet para encontrar lemas y eliminar palabras vacías como artículos, preposiciones, etc.

La siguiente fase del proceso consiste en gestionar las palabras analizadas y sus correspondientes funciones. Se ha optado por no emplear una matriz de coincidencias, donde se comparan entre si todas las palabras, y emplear una estructura dinámica que facilite técnicas de análisis de frecuencia. Las palabras provenientes de la ontología o de WordNet por vinculación de acepciones son almacenadas en una estructura formada por una tabla *hash* de grupos, donde cada entrada de la tabla corresponde a una palabra. La tabla crece dinámicamente a medida que se van analizando palabras, su estructura facilita el acceso y el inmediato conocimiento de coincidencias por nombre. A modo ilustrativo, siguiendo el ejemplo de la clase `arts` vemos una inserción de palabras en la tabla *hash* de grupos en la figura 5.1. Las palabras relacionadas con ella en WordNet se añaden a la tabla junto con su función semántica, es decir, el tipo de relación que guardan con respecto a la palabra. Las coincidencias no se solapan ya que cada elemento del hash es un grupo, y los elementos son únicos por función semántica. Por ejemplo, hay una coincidencia en el caso del término `art` que aparece como palabra simple y como palabra del significado (*gloss term*). La gestión de correspondencias se verá más adelante pero esta correspondencia es descartada por provenir el término de una acepción del mismo concepto. En este ejemplo, falta añadir las clases relacionadas en la ontología como son `activity` y `material`, junto con la propiedad `madeOf` más sus correspondientes relaciones semánticas que tengan en WordNet: acepciones, sinónimos, hiperónimos, etc.

Al explicar este pequeño caso vemos que el proceder del algoritmo es ineficiente en términos de información almacenada. La información se incrementa de manera exponencial dentro de la información definida en WordNet. Con este diseño el sistema colapsaría recursos e incluiría todas las palabras de WordNet. Si tenemos todo el vocabulario del tesauro cargado en memoria, éste hace referencia a todos los posibles contextos lo cual no resulta muy útil. Para evitar esto se han establecido una serie de criterios para finalizar este crecimiento en caso de encontrar un cierto patrón y acotando la inserción de elementos a ciertos niveles de profundidad.

Estos criterios se basan en dos casos. Uno donde la acepción de una palabra se averigua por el establecimiento de un patrón a través de la estructura de WordNet. En este caso las acepciones están relacionadas directamente. En el segundo caso, si este patrón no se da, la acepción es elegida a través de un conjunto de correspondencias ponderadas según la función semántica que ha

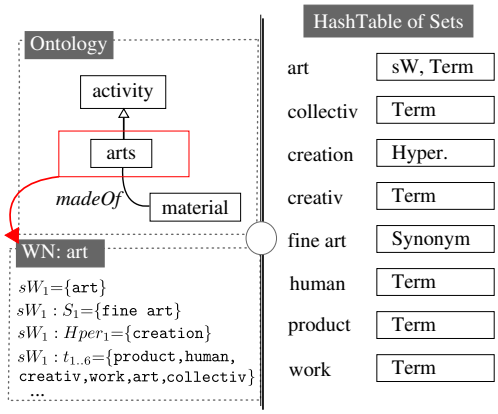


Figura 5.1: Información almacenada del concepto `art`

participado, es decir, en el peor de los casos hemos de recurrir al contexto de ambas representaciones.

Estos patrones relacionan directamente la acepción de dos clases mediante su correspondencia en WordNet. El patrón más evidente está producido por las relaciones jerárquicas que comparten ambas representaciones. Del ejemplo de la figura 5.1, la clase `arts` tiene una superclase `activity`, si se puede encontrar tal relación jerárquica en WordNet, ambos comparten un hilo significativo idéntico, entonces ambas tendrán sendas acepciones únicas. En la figura 5.2 vemos esta relación entre ambas acepciones. Este patrón determina las tres acepciones la de `arts` y `activity` conceptos de la ontología y la de `creation` concepto interno de WordNet.

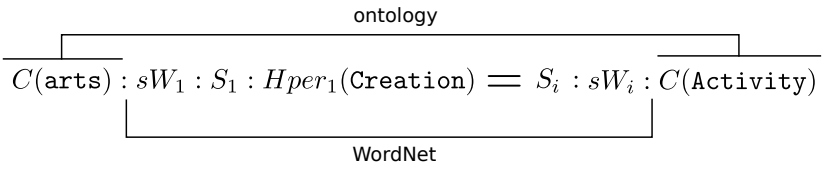


Figura 5.2: Coincidencia entre acepciones de `art` y `activity`

De las observaciones prácticas y de las relaciones entre los constructores de las ontologías y la estructura de WordNet hemos definido 4 patrones:

I) $C_i : sW_i : S_1 : \{Hper_i\} = sW_j : C_j$ y $C_i$ *owl:subClassOf* $C_j$ por relaciones jerárquicas de pertenencia. No es necesario que la relación de subclase sea directa, puede ser indirecta $C_i \subseteq C_k \subseteq C_j$. El ejemplo anterior corresponde a este patrón.

II) $C_i : sW_i : S_1 : \{Hpon_i\} = sW_j : C_j$ y $C_j$ *owl:subClassOf* $C_i$ por relaciones jerárquicas de subsunción. Al igual que el caso anterior la relación de subsunción no tiene porque ser directa.
III) $C_i : S_i : Syn_1 = C_j$ por relaciones de sinonimia.
IV) $C_i$ *owl:equivalentClass* $C_j$ y $C_{\{i|j\}} : |sW_1 : S_1| = 1$. Si dos clases son equivalentes y alguna de ellas posee una única acepción entonces ambas comparten la misma acepción y, por ende, el resto de términos asociados en la estructura y hechos de la ontología. Implica la propagación de esta nueva suposición entre todos los términos implicados y puede originar el cumplimiento de algún otro patrón.

La posibilidad de lanzar algunos de los patrones depende de la elección de términos por parte del diseñador y de su existencia en WordNet. El diseño de WordNet es el eje central por el que se identifican las acepciones y, por ende, los alineamientos. Por tanto, es evidente que los recursos externos afectan a la calidad de los alineamientos. Al hacer el símil con un proceso manual ocurre lo mismo, son nuestros conocimientos -nuestros recursos externos- los que posibilitan encontrar y definir los alineamientos.

En el caso de no aplicarse ningún patrón hemos definido un conjunto de correspondencias. Una correspondencia se produce cuando al menos dos palabras preprocesadas son iguales, es decir, hay una coincidencia en la tabla *hash*. La idea de la correspondencia surge porque habitualmente dentro de un dominio se utilizan palabras afines a ese dominio, posibilitando una mayor frecuencia de términos relacionados con el contexto. Por tanto, las acepciones más recurridas por coincidencia de términos serán la acepción con mayor aceptación dentro del concepto y de la propia representación. Las correspondencias son coincidencias léxicas entre diferentes conceptos y sus posibles términos derivados, y al menos ha de haber una acepción afectada. Es decir, la coincidencia de dos palabras simples no tiene ninguna repercusión pues no están vinculadas algunas de las acepciones.

A parte de valorar la correspondencia, se tiene en cuenta la función semántica del término en la relación, dependiendo si es un término de WordNet (sinónimos, hiperónimos, etc.) o de la ontología (subclase, equivalente, disjuntos, etc.)

El repertorio de correspondencias según la función semántica de los términos es el siguiente:

a) Casos de jerarquía:

$$C_i : sW_i : \{S_i : Hper_i | Hpon\}^r = \{*\} : S_j : sW_j : C_j$$

Son correspondencias por solapamiento de algún término de un concepto $j$ y con un término jerárquico del concepto $i$. Los patrones $I$ y $II$ prevalecen sobre este tipo de correspondencias. En ambas partes existe una acepción del concepto.
b) Propiedades entre clases. Las propiedades entre objetos van a cumplir dos funciones una causada por su firma (nombre, dominio y rango) y otra

por su tipo (transitiva, reflexiva, etc.). La firma de una propiedad es un elemento más para establecer correspondencias. El nombre es analizado como si fuera el de una clase y se derivan los correspondientes términos, se pueden aplicar patrones ya que sigue habiendo relaciones de jerarquía entre propiedades. Respecto a la función semántica sólo hemos valorado las propiedades transitivas por ser las que aportan una relación con otros conceptos y se puede utilizar para valorar acepciones y términos en común. Si se producen correspondencias causadas por una propiedad transitiva, los elementos implicados reciben las correspondencias aunque no tengan el mismo nombre. Por ejemplo, si tenemos una propiedad transitiva entre $a \rightarrow b$ y $b \rightarrow c$, y hay una correspondencia entre $a$-$b$ y $b$-$c$, entonces la correspondencia tendrá un mayor peso.

Hay un tipo de propiedades que contienen el término 'has' como primer token y han sido gestionadas de manera diferente. Son abundantes en la descripción de relaciones. En nuestros estudio (ver apéndice A) un 23 % de las propiedades contienen el término 'has'. Su uso es frecuente por la facilidad de asignar características y funcionalidades a las clases y, posiblemente, este incremento haya sido causado por el conocido manual de Protégé[2] que emplea este tipo de término. Como son utilizadas para definir cualidades estructurales o de pertenencia no es un término adecuado en nuestro sistema ya que no aporta algún tipo de relación al significado de la acepción de la relación o los conceptos implicados. Descartamos la partícula 'has' del nombre de la propiedad y el resto de elementos son tratados normalmente. Aunque la partícula 'has' no participe en la tabla, aquellas correspondencias por merónimos y holónimos reciben una valoración mayor ya que son interpretadas como funciones de pertenencia o posesión.

c) Propiedades de datos (*Datatype Properties*):

$$C_i : Pr_i : sW_i : S_i : \{*\} = \{*\} : S_j : sW_j : C_i.$$

Este tipo de correspondencias se consideran cuando las derivaciones por el nombre de la propiedad afectan al concepto vinculado en el dominio de la misma propiedad.

d) Restricciones. Como en el caso anterior, sólo se tiene en cuenta las correspondencias por el conjunto de las palabras derivadas de las propiedades en la restricción y cuando se vinculen conceptos implicados en la restricción.

e) Individuales. En caso de encontrarse en WordNet, generan correspondencias directas con aquellos conceptos de donde son tipo. Por ejemplo, mediante el análisis en WordNet del individual `Spain`, se proveen términos como `country`, `peninsula`, `iberia`, `europa`, etc. Estos términos pueden generar correspondencias con la clase a la que pertenece el individual `Spain`.

f) Resto de casos. Un total de 8x8 casos son posibles entre términos de la definición $t_i$, palabras simples $sW_i$, merónimos $Mer_i$, holónimos $Hol_i$, hi-

[2] http://owl.cs.manchester.ac.uk/tutorials/protegeowltutorial/

perónimos $Hper_i$, hipónimos $Hpon_i$, sinónimos $Syn_i$ y conceptos $C_i$. En este conjunto destacamos las más numerosas causadas por términos de la definición $sW_i : S_i : t_i{=}t_j : S_j : sW_j$, y las correspondencias entre sinónimos $sW_i : S_i : Syn_i{=}Syn_j : S_j : sW_j$.

A diferencia de los patrones, es habitual que se produzcan correspondencias entre diferentes acepciones del mismo concepto con lo que no se puede afirmar con rotundidad que acepción es la adecuada. Es decir, estamos ante la situación que el contexto influye en un determinado grado en diferentes acepciones. Además, en un 92 % de las correspondencias participan términos que no tienen una función semántica representativa entre acepciones. De ese 92 %, un 56 % son correspondencias entre términos de diferentes acepciones. Ver apéndice B.

Para decidir la acepción más adecuada, se ha optado por valorar cada una de las correspondencias según dos factores el tipo de correspondencia y la ambigüedad de los conceptos implicados.

La primera valoración, la valoración por tipo de correspondencia, se basa en la función semántica de ambos términos. Como hemos comentado anteriormente son correspondencias diferentes las formadas por términos y términos, que entre términos y sinónimos. Las primeras son más habituales y de escasa relevancia contextual. Los pesos según tipo de correspondencia están representados en la tabla 5.1. Hemos establecido tres conjuntos de pesos, los cuales son obtenidos experimentalmente y explicados en el apéndice C. La última fila, resto de casos (restricciones, individuales, etc.), presentan casos donde el peso no influye en la calidad. Por tener una misma función las siguientes categorías de merónimos, holónimos, hiperónimos e hipónimos son considerados como palabras simples aplicando en cada uno de ellos las restricciones oportunas comentadas anteriormente.

En la segunda valoración, se tiene en cuenta la participación en una correspondencia de un elemento no ambiguo, es decir, la palabra simple relacionada posee una única acepción. De esta manera, si en una correspondencia hay una acepción no ambigua, la acepción del otro concepto hace referencia a una interpretación concreta. Esta acepción tiene que ser más significativa que el resto de acepciones del concepto ambiguo. Esta valoración permite simplificar y reducir el análisis tal como comentamos en el trabajo relacionado (ver sección 3.7). Como ejemplo pongamos las siguientes correspondencias con términos del *gloss*, primera correspondencia $sW_a : S_a : t_a{=}t_x : S_x : sW_x$ donde $sW_a : |S_a| = 1$ y la segunda correspondencia con un clase $b$, $sW_b : S_b : t_b{=}t_x : S_x : sW_x$ donde $sW_b : |S_b| > 1$. La clase $a$ influye más por poseer una sola acepción que la clase $b$. Cuando un concepto tiene un único significado el peso de la correspondencia -por la primera valoración- se multiplica por 4, otro valor fijado experimentalmente en función de la calidad de los resultados.

| Tipo correspondencia | Peso *Simple* | Peso *Frecuencia* | Peso *Semántico* |
|---|---|---|---|
| término ∼ término | 1 | 6 | 1 |
| término ∼ palabra simple | 2 | 2 | 3 |
| término ∼ sinónimo | 2 | 2 | 3 |
| término ∼ propiedad | 1 | 1 | 1 |
| palabra simple ∼ palabra simple | 2 | 2 | 5 |
| sinónimo ∼ palabra simple | 2 | 1 | 5 |
| sinónimo ∼ propiedad | 2 | 1 | 5 |
| palabra simple ∼ propiedad | 2 | 1 | 5 |
| resto casos | 1 | 2 | 1 |

Cuadro 5.1: Tabla de valoraciones de las diferentes correspondencias, donde en cada relación se aplican las correspondientes restricciones, por lo general, el concepto ha de ser diferente en ambas partes.

## 5.4 Consideraciones y síntesis del contexto

Con el establecimiento de patrones y de correspondencias aún no se garantiza un crecimiento estable del número de términos. El rendimiento del algoritmo se ve afectado principalmente por el número de elementos recorridos en el recurso externo y al mismo tiempo, de aquellos que son almacenados para futuras comprobaciones.

Todas las palabras están relacionadas de una manera u otra en WordNet, por lo que a mayor número de palabras tratadas mayor es la probabilidad de establecer correspondencias. Si todas las palabras estuvieran delimitadas por algún tipo de información que diferenciara el dominio serían más prácticas, pero en el caso de WordNet no es así. En su diseño y en su estructura, la distancia de términos en función de las relaciones incrementa la incertidumbre en la elección de correspondencias. Es decir, cuanto más separados estén dos conceptos menos información específica tendrán y menos relevancia aportan a la descripción del contexto. Tal magnitud de palabras disipan el dominio. Por lo que no resulta útil gestionar todas las palabras.

Hemos ideado dos soluciones para delimitar el espacio de búsqueda. Una es reduciendo el número de relaciones y la otra es reduciendo la cantidad de clases a analizar. Estos dos planteamientos no han de menguar la calidad de los resultados, por tanto, las palabras y las clases seleccionadas han de seguir representando el contexto de la representación.

Para el primer caso, evitamos el crecimiento exponencial del espacio de búsqueda estableciendo un umbral que limite el análisis de las relaciones en WordNet. Este nivel de poda en las búsquedas unidireccionales de anchura (*breadth first search*) se ha fijado experimentalmente a una longitud máxima de 4 (ver apéndice D). Este valor se debe al propio diseño de WordNet tal como se explica en el apéndice.

### 5.4.1 Clases estructuralmente predominantes

El segundo planteamiento está basado en contemplar un número reducido de clases para establecer el contexto y a continuación, finalizar el proceso de descubrimiento con el resto de clases. Por tanto, hemos diseñado un proceso que analice las clases para determinar aquellas que son más significativas en la representación. Al delimitar el contexto con un número menor de clases se reduce el espacio de búsqueda. Esta reducción de recursos y síntesis de información es útil en sistemas de ambiente distribuido donde la ligereza de la aplicación es vital en sistemas con baja potencia y limitada energía [89]. La síntesis de elementos también es útil en sistemas de indexación o cacheado.

La idea se apoya en el sistema de valoración de acepciones donde las correspondencias con algún elemento no ambiguo hacen prácticamente seleccionable tal acepción y, también, en el hecho de que cuando a un concepto se le asigna una acepción, el resto de palabras relacionadas no vinculadas con la acepción se eliminan. Partiendo de estos dos hechos, al averiguar las acepciones de las clases estructuralmente predominantes y posteriormente las del resto, evitamos análisis indeseados. Estas clases permiten averiguar con mayor precisión la interpretación del resto de clases incluyéndose a si mismas.

Sintetizamos el contexto en un grupo de clases, llamadas clases estructuralmente predominantes (*structural predominant classes*, SPC). El diseñador interpreta y define un dominio donde involuntariamente hay elementos más característicos y, por tanto, predominan unos más que otros por una serie de factores. Dentro del posible repertorio de medidas extraíbles de una ontología, hemos seleccionado cinco. Estas cinco medidas son: (i) la profundidad relativa, es la mayor profundidad respecto a la profundidad de sus subclases, (ii) el número de subclases directas, (iii) el número de relaciones con rango sobre ella, (iv) el número de restricciones respecto a ella (v) y el número de individuales.

Son factores independientes por lo que una clase podría ser candidata por causa de varios factores. Vemos un ejemplo aplicado a la figura 5.3. Esta representado el esquema de una ontología con relaciones de jerarquías, propiedades, individuales -rectángulos-, y restricciones indicadas en el lado derecho. En la tabla 5.2 se muestra el valor de cada clase para cada criterio según el esquema anterior.

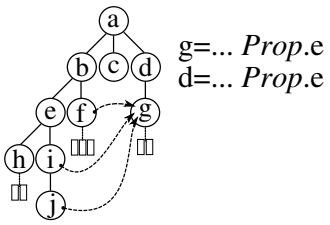


Figura 5.3: Boceto de la estructura de una ontología

| | clases | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **a** | **b** | **c** | **d** | **e** | **f** | **g** | **h** | **i** | **j** |
| Profundidad relativa | 5 | 4 | 1 | 2 | 3 | 1 | 1 | 1 | 2 | 1 |
| Subclases directas | 3 | 2 | 0 | 1 | 2 | 0 | 0 | 0 | 1 | 0 |
| Incidencias por rango | | | | | | 3 | | | | |
| Incidencias por restricciones | | | | 2 | | | | | | |
| Individuales | | | | | | 3 | 2 | 2 | | |

Cuadro 5.2: Criterios SPC bajo el esquema de la figura 5.3

Cuando se analiza una clase SPC no ignoramos las posibles relaciones con el resto de clases no marcadas como SPC. Si tiene relaciones con clases no SPC éstas se analizan. Así parte de ese contexto participa en la selección de la acepción de la clase SPC. El proceso de descubrimiento de acepciones se reestructura de la siguiente manera: primero se marcan las clases SPC y se descubren sus acepciones: por patrones o por correspondencias. Cuando finaliza la selección de acepciones, el algoritmo elimina toda aquella información no relacionada directamente con estas acepciones. A continuación se procede a descubrir las acepciones de los conceptos no marcados como SPC. Las clases se introducen en el sistema una a una y, antes de introducir otra se le asigna una acepción. Es decir, ya no habrá clases no SPC que aporten información contextual a otra clase de la cual no esté relacionada por algún patrón o correspondencia mediante una acepción innecesaria.

Con estos criterios todas las clases podrían ser etiquetadas como predominantes. Para evitarlo sólo se seleccionan aquellas con un percentil experimentalmente fijado a un 40 % para cada medida. Con esta idea se pretende no trasladar toda la información de la ontología a la estructura del algoritmo. Aún fijando este valor sigue existiendo la misma posibilidad. De todos modos, los experimentos han demostrado que el número de clases seleccionadas está alrededor del 30 % y del 40 % del total. Realmente puede mostrarse con las anteriores medidas la existencia de clases más significativas que el resto. Es recomendable ver apéndice E.

Como ejemplo, se visualiza la ontología *edas*[3] mediante dos distribuciones en la figura 5.4. En la sección de la izquierda ($a$), se ha representado mediante una distribución circular. Las clases son círculos y en un color más oscuro están marcadas las clases SCP. A simple vista, se aprecia que el mayor número de relaciones está presente en este conjunto. En la sección de la derecha ($b$) se ha optado por una distribución de fuerzas en grafos directos [59]. Se aprecia que los nodos más relacionados son los nodos centrales. En este tipo de visualización no están representadas los individuales y las restricciones. Por tal motivo es posible encontrar nodos aislados o en posiciones no tan "centrales". La visualización se ha realizado mediante un *plug-in*, llamado OWL-GEXF

[3] Una ontología del caso de estudio de conferencias de OAEI

parser, desarrollado especialmente para esta investigación. Transforma OWL en GEXF[4] y este formato es visualizado mediante Gephi [8].

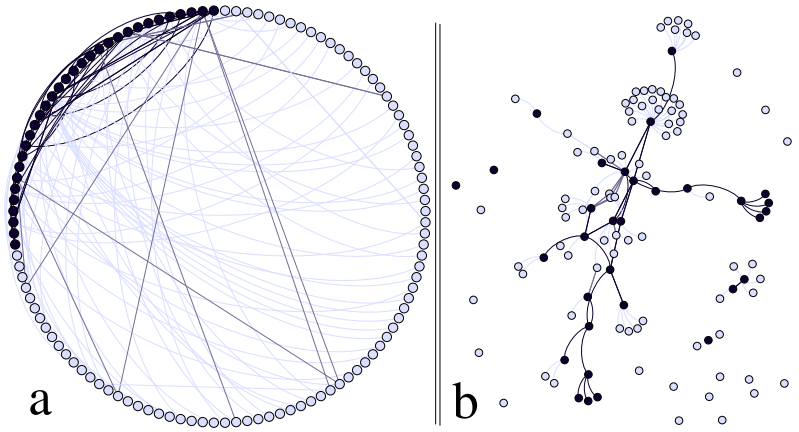


Figura 5.4: Dos distribuciones de la ontología *edas* con clases SPC marcadas

## 5.5 Nombres compuestos

Los nombres compuestos son un tipo de construcciones estructurales de regresivo asociamiento hacia la izquierda donde la adicción de un nuevo elemento inserta un nuevo asociamiento a todos los que le preceden [54]. Un nombre compuesto puede verse como una relación semántica entre una palabra referente (*head*) y los modificadores nominales. En el siguiente caso de nombre compuesto: `pizzaTopping`, `topping` es el referente y `pizza` es el modificador.

Los nombres compuestos son usados frecuentemente para aumentar la descripción de los elementos mediante el uso de varias palabras. Su uso es bastante extendido en la cultura anglosajona. En el caso de representaciones semánticas, el porcentaje de clases con nombres compuestos, sin tener en cuenta el resto de elementos, asciende a una media de un 56 % (ver apéndice F). Tal es su número que es necesario plantear su gestión para no mermar la calidad de los resultados. Sin embargo, hay tres propiedades que dificultan su interpretación: la composición es extremadamente productiva; la relación semántica entre el referente y los modificadores nominales es implícita y la interpretación puede estar influenciada por factores contextuales y pragmáticos [81].

Nuestra aproximación se basa en estudios llevados a cabo dentro del tratamiento de lenguajes naturales, especialmente en los trabajo de [80; 81; 82].

[4] http://gexf.net/format/

En [80] la obtención del significado de cada palabra se basa en técnicas de frecuencia bajo información extraída de WordNet. Las relaciones implícitas pueden ser clasificadas en: causa-efecto, contenido-contenedor, instrumento-agencia, origen-entidad, parte-todo, producto-productor y tema-herramienta. Usan WordNet para definir una fórmula de similitud basada en la similitud de los modificadores y referentes. En [132], las diferentes parejas se clasifican en 18 categorías para identificar la relación semántica entre ellas: *wrong parse, subtype, cause, characteristic, defect, attribute of clinical study, procedure,* etc. La identificación de cada pareja se basa en la creación de vectores con información jerárquica extraída de UMLS.

En nuestra suposición, la función semántica que desempeñan ambas palabras no es relevante para descubrir la acepción de cada palabra. Evitamos el proceso de identificación de la relación. En términos computacionales es complejo tener una única definición de palabras compuestas. Por ese motivo cada clase formada por nombres compuestos tiene múltiples acepciones, una por cada palabra. El descubrimiento de la acepción se basa principalmente en el establecimiento de una serie de casos evaluados en la representación de información en una ontología.

Para anticipar la exposición de casos de alineamiento, incluimos en este punto los supuestos en la equiparación de acepciones y similitudes entre referentes y modificadores. Con lo que el lector podrá comprender el papel de la acepción en los nombres compuestos.

(I) En el caso de que dos nombres compuestos coincidan por el referente consideramos:
- Si los modificadores son sinónimos, entonces ambas clases lo serán. Si fueran equivalentes su detección se haría en fases previas.
- Si ambos modificadores comparten una relación de jerarquía entonces ambas clases serán subclases una respecto a la otra.
- Si ambos modificadores comparten un sujeto similar (p. ej. *orangeJuice* y *appleJuice*), entonces se crea una clase etiquetada con el referente donde las dos clases son subclases de esta (p. ej. *orangeJuice* $\subseteq$ *Juice* y *appleJuice* $\subseteq$ *Juice*). El sujeto similar es detectado por compartir o poseer un predecesor similar (p. ej. en este caso es *fruit*). De esta manera, se evita reestructurar la representación del diseñador.

(II) Existe la posibilidad de relacionar un referente de una palabra con un modificador de la otra pero es imposible determinar el tipo de relación por la no existencia de recursos. Cada caso se debe tratar individualmente siendo un proceso asistido. Por este último motivo, no lo hemos considerado.

(III) En el caso que ambos modificadores coincidan, p. ej. `orangeJuice` y `orangeFruit` o `freshJuice` y `freshFruit` tampoco podemos determinar el tipo de relación que puede haber entre ellas. Vuelve a ser un proceso asistido y tampoco lo hemos considerado.

En resumen, el significado de una palabra compuesta es difícil de representar pues todos los modificadores determinan un dominio específico o una cualidad del referente. Hemos considerado tratarlas como unidades independientes con la finalidad de determinar la equivalencia exacta por acepción durante el proceso de mapeado. Finalmente, introducimos una serie de casos de alineamientos que serán detallados en profundidad en la sección 7.2.

## 5.6 Conclusión

El descubrimiento del significado de elementos de una ontología es un proceso complejo e impreciso en muchas ocasiones. El proceso sigue algunas ideas para la gestión de datos, obtención de términos y análisis de recursos externos de trabajos relacionados en la manipulación de lenguajes naturales. Técnicas basadas en frecuencias, en vectores, son relativamente imprecisas y hemos definido una serie de reglas: patrones y correspondencias. Estás reglas son comparaciones de las relaciones y constructores de WordNet con las ontologías, respectivamente. Los patrones relacionan acepciones directas donde no hay posibilidad de incertidumbre. En cambio, las correspondencias sí contemplan la frecuencia de términos pero estos son identificados por su función semántica extraída de WordNet y de la ontología. Es una tupla de datos más útil que una simple palabra. Finalizado el proceso de búsqueda de términos para establecer estas reglas, se determinan las acepciones de todas las clases. Y ésta es almacenada en la propia ontología para futuras tareas.

A la hora de utilizar recursos externos, el problema es acotar el espacio de búsqueda para que el algoritmo no contemple todos los posibles términos. Como todos los términos están relacionados, esto no aporta ninguna utilidad para delimitar aquellas palabras que forman parte del dominio de la representación. Hemos reducido el espacio de búsqueda mediante el establecimiento de umbrales fijados experimentalmente sin que afecten a la calidad de las acepciones. Al mismo tiempo, descubrimos que existe un número determinado de clases que por razones de diseño son más representativas que el resto, en función de unos criterios. La gestión anticipada de estas clases reduce el uso de recursos y tampoco afecta a la calidad.

Otro punto a tratar es la identificación de la acepción de palabras compuestas. Es un tema tratado en numerosos trabajos previos. Hemos optado por tratar cada palabra de manera independiente: anotando su acepción. En el proceso de alineamiento, la relación entre el referente, los modificadores y sus acepciones sí permitirá fijar los correspondientes alineamientos.

El proceso de descubrimiento del significado contempla procesos que se realizan en cualquier algoritmo de mapeado por lo que es un trabajo reutilizable. El preprocesado lingüístico, el análisis estructural, la reducción del espacio de búsqueda queda reflejado en las estructuras basadas en *hash* y grupos utilizadas en este proceso. Estas estructuras algebraicas son comparables entre sí, lo que facilita la equiparación de ontologías y por tanto, su mapeado.

# 6

# La representación de los alineamientos

La representación de los alineamientos desempeña un papel importante a la hora de manipularlos, reusarlos o integrarlos en otras aplicaciones sin tener que recurrir a una aplicación independiente fuera del repertorio habitual de desarrollo e influye directamente en el proceso de descubrimiento de alineamientos. Por eso motivo, proponemos la definición de nuevas relaciones en OWL que permitan representar alineamientos con el objetivo de compatibilizar al máximo los resultados con cualquier API usual en el manejo de ontologías. Esta propuesta es llamada OWL-M, 'M' proviene de mapeado y el acrónimo `owlm` define el prefijo en la importación del vocabulario[1]. Con las nuevas relaciones definidas en OWL-M se aumenta la interpretación de los alineamientos, disminuyendo su ambigüedad. Además, se han definido un conjunto de anotaciones para identificar los algoritmos y las técnicas que tienen parte en cada alineamiento. Finalmente, sobre OWL-M se ha definido un sistema para calcular la probabilidad de confianza sobre el alineamiento en aquellas situaciones donde sintácticamente se necesite este valor numérico. Desde nuestro punto de vista, esta probabilidad la hemos denominado distancia conceptual entre dos clases por tener una aplicación no limitada al mapeado de ontologías [90].

Un alineamiento es una relación semántica entre dos conceptos que puede verse como un camino con origen y final entre las clases implicadas. Este camino habitualmente contiene dos clases y una relación. Tener una sola relación limita la comparación de clases a aquellas que están separadas por varias relaciones. Para evitarlo, suponemos que los alineamientos pueden contener múltiples clases y varias relaciones, a lo que llamaremos camino compuesto. De esta manera, se flexibiliza y se enriquece el repertorio de alineamientos. Aunque por otro lado, la dificultad de descubrir enlaces entre clases distantes se hace cada vez mayor. Tal análisis se desarrolla en dos partes según el tipo de camino simple o compuesto, alineamiento simple o compuesto respectivamente.

[1] http://swap.uib.es/2009/08/owlm

## 6.1 Alineamientos simples

Un alineamiento simple es una la relación directa entre dos clases. Bajo esta definición los siguientes constructores forman alineamientos simples.

- Relaciones de jerarquía por subclases y por superclases se han nombrado de manera diferente a OWL (`rdfs:subClassOf`) como `owlm:Hyponym` y `owl:Hyperonym` respectivamente. La distinción permite diferenciar la procedencia del constructor dentro de la misma representación. El modo lógico subyacente es el mismo.
- Relaciones de equivalencia, `owlm:equivalent`, cuando dos clases comparten la misma definición. Tiene el mismo modelo lógico que la relación de OWL `owl:equivalentClass`. El objetivo es diferenciar la procedencia del constructor en la representación.
- Relaciones de sinonimia, `owlm:synonym`, cuando dos clases comparten mismas características, intercambiables en el mismo contexto pero tienen matices diferentes. A nivel lógico la propiedad se considera como una equivalencia pero el constructor tiene un matiz diferente, donde en ciertos contextos este intercambio no tendría sentido, depende de la naturaleza de la aplicación final.
- Relaciones de antonimia (`owlm:antonym`, $\perp$) es una relación basada en la complementariedad de un concepto. Podemos diferenciar diferentes tipos de relación: reciproca (comprar $\perp$ vender), complementaria (vida $\perp$ muerte) y gradual (negro $\perp$ blanco, frío $\perp$ calor). Para definir una relación de este tipo es necesario que haya un hiperónimo en común, lo que habilita un entorno común de características, por ejemplo {color**:**negro $\perp$ blanco}; {temperatura**:**frío $\perp$ calor}; con superclases diferentes no podría establecerse tal relación entre por ejemplo {*illness:cold*} $\pm$ {*temperature:hot*}. El tipo de relación es mediante una de las tres siguientes anotaciones: `owlm:reciprocal`, `owlm:complementary` y `owlm:gradual`.

Estos cinco constructores son axiomas de clase ya que restringen la definición del concepto. Su representación formal se muestra en las primeras cuatro filas de la tabla 6.1. La segunda columna es la representación mediante sintaxis abstracta usando *OWL Abstract Syntax Style* (siguiendo la nomenclatura recomendada por W3C). La siguiente columna representa la sintaxis de lógica descriptiva. La cuarta columna contiene la representación semántica basada en el artículo de [57] y la recomendación del W3C. La última columna contiene algunos ejemplos usando símbolos matemáticos adecuados e interpretables.

Los anteriores constructores son los más significativos para realizar integraciones de información como por ejemplo equivalencias de bases de datos, en *queries*, en servicios, en parámetros, etc. Al trabajar con recursos externos con información semántica, sean tesauros o ontologías, incrementamos significativamente el contexto de una clase por una mayor aparición de propiedades con significado. Nos encontramos con posibles relaciones entre elementos más

distantes semánticamente que los anteriores constructores. La nueva información contextual mejora la integración y las búsquedas. Para formalizar la representación de este tipo de caminos entre contextos, hemos definido las siguientes propiedades:

(i) `owlm:partOf` permite definir la composición de objetos. Este constructor determina la pertenencia de un objeto a una organización particular con propiedades o funcionalidades comunes bajo la estructura del mismo. Por ejemplo, supongamos una relación donde un ordenador portátil `tiene` un teclado y un monitor. Esta relación implica algún tipo de organización estructural entre los componentes: para que un portátil desempeñe su función necesita del teclado y del monitor. Otro caso es la relación de pertenencia o de posesión, p. ej. una persona `tiene` un perro. Este último caso, los elementos implicados no definen al primer objeto, el perro no define a la persona ni su función. La relación *partOf* es transitiva, si un elemento $A$ es *partOf* $B$ y $B$ *partOf* $C$ entonces $A$ *partOf* $C$.

(ii) `owlm:composedOf` permite la descomposición de un objeto en otros. Es una relación inversa a *PartOf*. Tanto `owlm:partOf` como `owlm:composedOf` son relaciones que pueden aplicarse junto con restricciones de cardinalidad. Lo cual resulta útil porque son numerosos los casos donde existen composiciones donde participan más de un elemento. Por ejemplo, un teclado *composedOf* 103 teclas.

(iii) *Relation-Action* ($\overline{RA}$ ), su definición es $C \xrightarrow{R} D$. Donde, $R$ es un verbo que representa la acción realizada por la clase $C$ usando $D$, dominio y rango respectivamente. La acción puede llevarse a cabo en $C$ sin la intervención de un rango, siendo $D$ un elemento vacío. O también, se puede definir sin una relación explícita $R$ propiciando una regla de implicación de primer orden sin clausulas de Horn. Cuando se define una $\overline{RA}$ al menos uno de los dos elementos, $R$ o $D$, deben definirse. Por otro lado, $R$ puede contener un descriptor del lugar donde se realiza la acción ($R_{where}$). Si $R$ no se define entonces $R_{where}$ tampoco. (iv) Con $\overline{RA}$ se pueden representar verbos intransitivos donde el objeto no existe en la interpretación ($\Delta^I$) con $D = \emptyset$, por ejemplo $Person \xrightarrow{run} \emptyset$. (v) Los constructores `owlm:composedOf` y `owlm:partOf` pueden ser interpretados como un tipo de $\overline{RA}$ , donde el verbo $R$ puede ser[2] o "*to compose*" o "*to form*" respectivamente, por ejemplo: $Pizza \xrightarrow{compose} Bread$ y $Camshaft \xrightarrow{form} Engine$. Bajo otra consideración, podemos poner otro ejemplo usando un verbo sinónimo: $Balloon \xrightarrow{contains} Air$. La diferenciación explícita entre ambos se debe a la facilidad de interpretación y uso de los dos primeros frente a una $\overline{RA}$ . (vi) Siempre que la acción sea diferente un concepto puede tener múltiples $\overline{RA}$ . Además, esta propiedad puede ser definida como transitiva, reflexiva y funcional y poseer inversa. Para clarificar la idea de $\overline{RA}$ mostramos una serie de ejemplos: $Person \xrightarrow{eat} Pizza$, $Car \xrightarrow{consume} Gasoline$, en ambos casos el lugar es indiferente a la acción. En la figura 6.1 incluimos una serie de

[2] bajo nuestra interpretación

ejemplos: (a) un suicidio implica muerte, el lugar de la acción es indiferente; (b) una persona bebe líquido potable, dando igual el lugar; (c) un avión vuela, no importa el lugar pero podríamos especificar el aire, el cielo, la atmósfera, etc.; (d) un bebe duerme en una cuna.

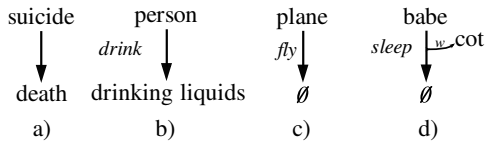


Figura 6.1: Ejemplos de reglas de acción

La diferencia entre una relación $\overline{RA}$ y una `owl:ObjectProperty` estriba en el origen de la afirmación. Si el diseñador es quién decide interpretar el dominio de esta manera, dispone de reglas y mecanismos para representar una $\overline{RA}$ con los constructores de OWL. En cambio, cuando esta implicación es causada por mecanismos de interpretación externos, como en nuestro caso un algoritmo de mapeado, es aconsejable hacer la diferenciación usando un constructor diferente donde quede reflejado el objeto y el lugar de la acción. Con este constructor enriquecemos la interpretación de las clases aumentando la posibilidad de establecer alineamientos sobre elementos de otros dominios y enriquecemos la información sin alterar el esquema original. Esta separación permite mantener la estructura original del modelo.

Una manera alternativa de indicar esta información puede ser mediante el uso de anotaciones. Sin embargo, las anotaciones, a diferencia de un constructor, no pueden ser utilizadas directamente por el modelo lógico si no por aplicaciones o reglas lógicas específicas. Esto va en contra del objetivo de utilidad de las representaciones semánticas.

En esta investigación no se ha desarrollado ningún razonador capaz de tratar con estos constructores. En la práctica hemos definido una serie de reglas para operar y mostrar la utilidad de los constructores de OWL-M.

| Name | Abstract Syntax | DL Syntax | Semantics | i.e (under $\Delta^I$) |
|---|---|---|---|---|
| Class Axioms | | | | |
| Hyponym | Hyponym($C$ $D$) | $C \sqsubseteq D$ | $C^I \subseteq D^I$ | $Dog \subseteq Canido$ |
| Hyperonym | Hyperonym($C$ $D$) | $C \sqsupseteq D$ | $C^I \supseteq D^I$ | $Vehicle \supseteq Truck$ |
| Equivalence | Equivalence ($C_1 ... C_n$) | $C_1 \equiv ... \equiv C_n$ | $C_1^I \equiv ... \equiv C_n^I$ | $City \equiv City$ |
| Synonym | Synonym($C_1 ... C_n$) | $C_1 \equiv ... \equiv C_n$ | $C_1^I \equiv ... \equiv C_n^I$ | $Hot \equiv Torrid$ |
| Antonym | Antonym($E : C$ $D$) | | $C^I \cap D^I = \emptyset$ and $\exists E^I \mid C^I, D^I \subseteq E^I$ | $Temp. : Hot \perp Cold$ |
| Cause | Cause($C$ $D$) | $C \rightarrow D$ | if $C^I$ then $D^I$ | $fire \rightarrow hot$ |
| | $res.(R\ repetition(n))$ | | $R^I$ *is* $PartOf, m = n$ | |
| Properties | | | | |
| PartOf | $[PartOf]$ | | $\exists C^I \equiv \{D_i^I\}^{*m} - D^I = \{D_1^I, D_2^I, ..., D_m^I\}, m = 1, R^I \subseteq C^I \times D^I$ | (1)$Camshaft\ belong\ Engine$ |
| ComposedOf | $[ComposedOf]$ | | $R^I = (\ ^- PartOf_0^I)$ | $Pizza\ has\ 1.Bread$ |
| ActionRule | ActionRule(R E) | | $R^I \subseteq C^I \times D^I$, where $C^I, D^I$ occurs at $E^I$ | $fan \xrightarrow[where:Earth]{generate} wind$ |

Cuadro 6.1: Sintaxis abstracta y semántica de los constructores de OWL-M

La sintaxis abstracta de cada uno de los constructores se puede ver en la tabla 6.2. El ejemplo 6.1 muestra un uso puntual.

| Abstract Syntax | Transformation |
|---|---|
| *<ID>* | *<ID>* |
| *Class(<classID>...* | *<classID> rdf:type owl:Class .* |
| Hyponym($C$ $D$) | [<classID> owlm:hyponymOf :_x .] |
| Hyperonym($C$ $D$) | [<classID> owlm:hyperonymOf :_x .] |
| Synonym($C_1 ... C_n$) | [<classID> owlm:synonymOf :_x .] |
| Antonym($E : C$ $D$) | [<classID> owlm:antonymOf :_x .] |
| Cause($C$ $D$) | [<classID> owlm:causeOf :_x .] |
| *ObjectProperty(<name> ...* | *<name> rdf:type owl:ObjectProperty .* |
| [$PartOf$] | [<name> rdf:type owlm:partOf . ] |
| [$ComposedOf$] | [<name> rdf:type owlm:composedOf . ] |
| Action(R in C) | <name> rdf:type owlm:actionRule .<br>[<name> owlm:whereAR <classID> .]<br><name> rdfs:domain T(<domain1>) ...<br><name> rdfs:domain T(<domainN>) .<br>[<name> rdfs:range T(<range1>) ...<br><name> rdfs:range T(<rangeN>)] . |

Cuadro 6.2: Sintaxis abstracta para la transformación a RDF

Código 6.1: **Ejemplo de sintaxis abstracta**

```
Class(SpanishSandwich
  partial Sandwich
  restriction (hasBread minCardinality(2)))
  ObjectProperty(hasBread
     domain(Sandwich)
     range(class (BreadBaguette partial Bread))
     ComposedOf)
}
```

## 6.2 Alineamientos compuestos

Algunos casos de similitud requieren de combinaciones de constructores simples. Un claro ejemplo son las relaciones de jerarquía. Hay una cierta relación de similitud entre una subclase y la superclase de una clase ($x \sqsubseteq y \sqsubseteq z$) y lo mismo puede ocurrir con diferentes combinaciones de constructores ($x \sqsubseteq y \sqsubseteq z \equiv h$). Desde el punto de vista de la integración de datos puede no resultar útil ya que no relaciona dos conceptos directamente por una relación simple. De todos modos, este tipo de composiciones resulta útil para establecer agrupamientos de elementos afines entre si o a un dominio, definir restricciones mediante una lógica más rica en cuanto constructores y para el descubrimiento de servicios entre proveedores comunes.

Vamos a suponer una serie de casos para ilustrar este tipo de alineamientos compuestos donde relacionamos la clase *globo* (*balloon*) y la clase *agua* (*water*). En la figura 6.2 se han establecido tres alineamientos -caminos- diferentes. El camino $A$, tal vez el más complejo y difícil de establecer por un sistema computacional, determina la relación mediante la combinación de una subclase de *globo* la cual es un tipo de *globo aerostático*, un tipo de *vehículo aéreo*, que usa el *aire* como un *medio de transporte*. Desde el concepto de *agua*, el *agua* es originada por la *lluvia* la cual produce *nubes* que comparten el mismo ambiente que los *globos aerostáticos*. En el camino $B$ se interpreta que el *globo* está compuesto de *aire* donde comparten el espacio con las *nubes*.... El último camino supuesto, el $C$, puede ser el más flexible pero nada trivial. Un *globo* contiene *gas*, el cual es un antónimo de *líquido*. El *líquido* tiene un tipo de instancia: *agua*. Todas estas relaciones pueden ser difíciles de establecer por un sistema inteligente que disponga de todo este conocimiento. Sin embargo, la relación entre los conceptos existe por lo que se ha de poder definir y también computar.

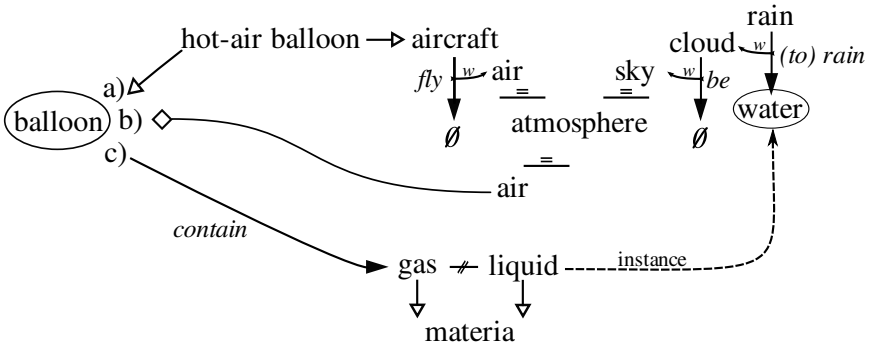


Figura 6.2: Tres maneras de relacionar los conceptos de *balloon* y *water*

## 6.3 Anotaciones para describir el método de alineamiento

Los anteriores constructores permiten definir una relación de similitud entre conceptos pero en ellos no se vincula quien lo ha realizado, que información ha utilizado y que tipo de estrategia se ha utilizado. Se hace necesario disponer de información descriptiva de la relación para facilitar tareas de documentación, estadísticas y reutilización de los alineamientos producidos. Para tal fin,

se han definido un conjunto de anotaciones donde se definen los siguientes datos: autor, versión, comentarios del autor, conjunto de algoritmos utilizados, precisión numérica y estrategia. La estrategia es un conjunto de etiquetas que permiten clasificar el proceso entero. De esta manera, se representan características tales como: restricciones, lenguaje, comparativa de cadenas de texto, nivel estructural, reuso de alineamientos, entre otras posibilidades.

## 6.4 Cálculo del umbral de semejanza

A lo largo de este trabajo hemos defendido la representación semántica mediante constructores lógicos de los alineamientos como requerimiento fundamental para disponer de un algoritmo semántico. Sin embargo, a nivel computacional disponer de un valor numérico de similitud entre conceptos es eficiente y simplifica numerosos procesos. Por tal razón hemos optado por plantear un método para calcular este valor de similitud en función del camino.

Hay diferentes estudios de como computarlo acorde a la taxonomía de WordNet [16]. Una de las aproximaciones más eficientes está basada en la longitud del camino. Según Resnik: "Dos conceptos serán más similares si el camino entre ellos es corto" [130].

Hemos propuesto calcular este valor en base a dos tipos de medidas: el camino y el contexto. La información que podemos extraer a partir del camino son los tipos de constructores y la longitud. El camino no es único, puede variar entre diferentes esquemas y mismos conceptos ya que depende del algoritmo, recursos externos y del contexto de la representación. Por otro lado, a partir del contexto disponemos de caminos hacia las clases de la propia representación, lo cual provee un valor sobre la similitud de ambas representaciones desde cada una de las clases implicadas. La ponderación depende de la distancia de estas clases hacia la clase central, siendo menos relevante a medida que se distancie del origen. En resumen, los resultados son variables porque dependen del contexto, así por ejemplo, dos clases como *car* y *motor vehicle* tendrán un grado de similitud en función de la similitud del contexto

La fórmula propuesta (6.1) se divide en dos partes: un cálculo de la similitud por contexto ($P_{Cx_1,Cx_2}$) y el peso por el camino ($\sum weight(r_i)/len(c_1, c_2)$). En la similitud por contexto se calcula el número de palabras iguales a una distancia $i$ por el total de elementos a esa distancia. La distancia está fijada experimentalmente a 4 por ser el número óptimo de profundidad donde se establecen la mayoría de las relaciones (ver apéndice D).

La valoración del camino es mediante la división de una ponderación constante de cada constructor implicado $weight(r_i)$ en el camino y de la longitud del mismo $len(c_1, c_2)$. Los pesos de cada constructor se han fijado experimentalmente (ver tabla 6.3 de ponderaciones). Este peso varía en función de los objetivos del mapeado, es decir, podemos pretender encontrar equivalencias exactas, diferencias mediante antónimos, o cualquier otra combinación deseada. Es la propia naturaleza de la aplicación final, que requiere de estos

resultados, la que valorará unos resultados más ajustados a su finalidad que otros. Los pesos mostrados en la figura están fijados para promocionar las relaciones de sinonimia sobre el resto, fomentando alineamientos para esquemas de datos.

$$sim(c_1 \xrightarrow{r} c_2) = P_{Cx_1,Cx_2} \frac{\sum weight(r_i)}{len(c_1, c_2)}$$
$$P_{Cx_1,Cx_2} = \sum w_i * Elements_i / TotalElements_i \tag{6.1}$$

| Constructor | Peso |
|---|---|
| `synonym` | 4 |
| `antonym` | 0 |
| `hyperonym` | 2 |
| `hyponym` | 2 |
| `part` | 1 |
| `composed` | 1 |
| `action` | 2 |
| `where` | 3 |
| `instance` | 4 |

Cuadro 6.3: Pesos experimentales asociados a cada constructor

## 6.5 Conclusión

En este capítulo, hemos presentado una serie de constructores basados en OWL para representar diferentes relaciones semánticas entre conceptos a la hora de equipararlos. Hemos llamado OWL-M a este conjunto de constructores. El lenguaje es totalmente compatible con OWL por lo que puede ser utilizado por las mismas librerías de programación y razonadores de inferencia. Por tanto, no se requiere de mecanismos externos supervisados o no que determinen la validez de una relación con un valor numérico. En OWL-M se definen una serie de anotaciones útiles para facilitar y documentar la identificación de acepciones y el proceso de mapeado, potenciamos el estudio y análisis de técnicas estadísticas sobre las operaciones llevadas a cabo en cada uno de los alineamientos. OWL-M no es más que una aplicación de OWL a un domino específico, pero hemos defendido la necesidad de utilizar un estándar de representación de alineamientos que proporcione información más relevante que un valor numérico y una relación semántica indeterminada.

# 7

# Algoritmo OMoCC

Como ya hemos mencionado OMoCC no es más que una herramienta donde se aplican las ideas principales de esta tesis con el fin de demostrar su viabilidad e impacto. El algoritmo OMoCC integra el proceso de descubrimiento de acepciones por lo que reutiliza una estructura de datos diseñada específicamente para establecer similitudes entre palabras con sus respectivas funciones semánticas dentro de WordNet y la propia ontología. Al usar esta estructura evita realizar un preprocesado lingüístico de los elementos junto con procesos secundarios a la ahora de realizar el proceso de mapeado. Esta estructura contiene las correspondencias por nombre entre los elementos de ambos esquemas lo cual facilita el descubrimiento de alineamientos. Como representamos los alineamientos mediante OWL-M, el proceso de búsqueda de los mismos se enfoca en el repertorio de constructores disponibles en OWL-M.

## 7.1 Proceso de desambiguación

Con el objetivo de detallar el funcionamiento de OMoCC, primero explicamos el proceso de desambiguación de acepciones con pseudo-código. No hemos entrado en detalles de implementación innecesarios para la compresión de ciertas funciones o consideraciones de las variables de las clases. En el código 7.1 adjuntamos el procedimiento del algoritmo de desambiguación. La sintaxis de éste y de los siguientes pseudo-códigos está basada en Java, donde los constructores y variables están en inglés. Además, sintetizamos fases y funciones en comentarios para facilitar su comprensión y simplificación del volumen de datos.

Código 7.1: **Pseudo código del proceso desambiguación**

```
MeaningDiscovering(..):
//Carga API WordNet
//Carga algoritmo de analizador de Porter
for (Ontology onto: listMappingOntologies){
  //Carga del modelo de inferencia
  OntModel model = getInferenceModel(onto);

  //Determinar clases estructuralmente predominantes
  ArrayList<URI> listSPC = getStructuralPredominantClasses(model);

  //Se averiguan las acepciones de las clases SPC
  for (Concept concept: listSPC)
    addConcept(concept); //función comentada.

  //Las acepciones de todas las clases SPC se fijan
  //en función de los patrones y correspondencias
  //El resto de palabras no involucrados son eliminadas

  //Una vez determinado las acepciones de las clases SPC
  //se determina una a una la acepción de cada clase NO SPC
  for (Concept concept: listNoSPC){
      addConcept(concept);
      fijarAcepción(concept);
  }
  //Las anotaciones son guardadas en la ontología mediante  OWL-M
/**
 * En addConcept() se gestionan los patrones, correspondencias  y
 * se insertan los elementos vinculados a la palabra en la tabla de Hash
 * Si se produce un patrón, la acepción es establecida,
 * el proceso se detienen sólo se insertan y analizan las palabras
     relacionadas
 * con la acepción establecida.
 */
addConcept(..):
    //¿Se ha procesado anteriormente?
    //¿Tiene significado?
    if (!concept.hasMeaning()){
      //Preprocesado lingüístico: camelcase, tokens, números, etc.
      //Analizador de Porter
      if (!concept.hasMeaning() && concept.isCompoundWord()){
        //Análisis estructural del conjunto con WordNet
      }else{
        //Análisis jerárquico: nuevos conceptos con WordNet
      }
  }
  //Comparando elementos con WordNet:
  // - Análisis de propiedades
  // - Análisis de individuales
  // - Análisis del Gloss
```

No exponemos las funciones de análisis en palabras compuestas, jerárquicas, propiedades, individuales y términos del *gloss*. En todas estas fases, las palabras provenientes de WordNet pasan por el preprocesado lingüístico donde se eliminan las palabras innecesarias. Aquellas palabras que generen correspondencias o patrones determinarán la posible acepción y son almacenadas en la estructura *hash* con su correspondiente información de tipo, acceso y lugar de procedencia -clase o palabra simple de donde ha surgido-. Por poner un ejemplo, el análisis estructural o jerárquico busca el establecimiento de patrones y correspondencias. Coteja ambas estructuras en busca de alguna coincidencia y dispara todos los procesos de almacenamiento y tratamiento

de los mismos. Como la ventana o espacio de búsqueda está limitado por cuestiones de eficiencia (máxima profundidad y análisis de caminos) durante el análisis de otros elementos podrían aparecer palabras ya analizadas y/o guardadas. La aparición de nuevos elementos no genera ningún tipo de problema, ya que la tabla *hash* guardaría los nuevos lugares de procedencia y, en ciertas ocasiones, actualizaría este lugar según la importancia de la fuente dando prioridad a las propias ontologías en vez de a WordNet. Este mismo hecho puede producirse fácilmente con el análisis del *gloss*. En la definición de elementos de una ontología es recurrente referenciar a otros elementos. La gestión de la recurrencia de términos ha sido una de las tareas más complejas pero a su vez gracias a la estructura lógica de tablas *hash* y grupos ha sido fácil de implementar.

Cuando se analizan los patrones y las correspondencias de las clases SPC, se selecciona una acepción. Los identificadores de esta acepción son almacenados en la propia ontología (o en otra temporal) mediante anotaciones de OWL-M. Cuando finaliza la anotación del significado de las clases SPC se determina la acepción de todas las clases SPC en función de las correspondencias. A partir de ese momento se eliminan las palabras no relacionadas con esas acepciones. Posteriormente, se introducen una a una las clases no SPC y al instante se fija su acepción y se eliminan las acepciones y palabras relacionadas no elegidas. El proceso continúa hasta analizar la última clase no SPC. Una vez finalizado este proceso de desambiguación conocemos el significado de cada clase y en la estructura de datos disponemos de todas aquellas palabras relacionadas con las acepciones descubiertas.

## 7.2 El proceso de descubrir alineamientos

El proceso determina la similitud en función de dos grupos de medidas disponibles: la detección de equivalencias por significado y el establecimiento de relaciones por otro conjunto de correspondencias averiguadas en la fase previa de desambiguación. Al ser los resultados representados mediante OWL-M, se ha optado por incluirlos en un modelo a parte ya que los alineamientos suponen una extensión del modelo lógico de las representaciones originales.

La primera medida es la más sencilla por el trabajo realizado durante el proceso de desambiguación. El algoritmo ha de encontrar aquellas clases entre ambas estructuras que tengan la misma acepción. Estos alineamientos se definen como `owlm:equivalentOf`. Independientemente del nombre de la clase, de su estructura y de las inferencias que tengan en sus respectivos modelos, ambas serán equivalentes por poseer la misma acepción. Lo mismo ocurre con palabras compuestas con significado propio, pues son tratadas como palabras simples. Se puede comprender que la calidad de los alineamientos depende directamente de la calidad del algoritmo de desambiguación.

Como una clase puede estar compuesta de diferentes palabras (palabras simples), estas pueden dar lugar a un conjunto de casos que se han de tratar

mediante una serie de reglas. Estas reglas fueron explicadas en la sección 5.5 pero son detalladas en esta sección:

- *Regla 1.* Si dos nombres compuestos comparten el mismo nombre entonces son dos clases equivalentes independientemente de las acepciones. Rara vez existen conceptos compuestos iguales con significados distintos. Las múltiples palabras fortalecen la restricción sobre la interpretación global en el dominio de la acepción. Por eso, no resulta necesario comparar acepciones. Son escasas las excepciones, y además en esos pocos casos es habitual que una de las palabras tenga significado por sí sola, por ejemplo: `AirForce` -avión presidencial-, `Air force` -flota de aviones militares-.
- *Regla 2.* Cuando los referentes de dos palabras compuestas coinciden ocurren una serie de casos:
  - *Regla 2.1.* Si los modificadores son sinónimos entonces ambas clases también lo serán.
  - *Regla 2.2 / 2.3.* Si los modificadores presentan una relación de hiperónimia/homónimia en WordNet uno respecto al otro, serán superclase/subclase respectivamente.
  - *Regla 2.4.* Si los modificadores comparten una superclase en común, entonces, en caso de no existencia, se crea una clase con el significado del referente, siendo esta clase superclase de ambas palabras compuestas.
- *Regla 3.* Cuando el referente de una palabra compuesta coincide con un concepto simple, la palabra compuesta se transforma en una subclase de la palabra simple. Es decir, la palabra compuesta es un tipo específico del referente. Por ejemplo, *PizzaTopping* se transforma en una subclase de *topping*.

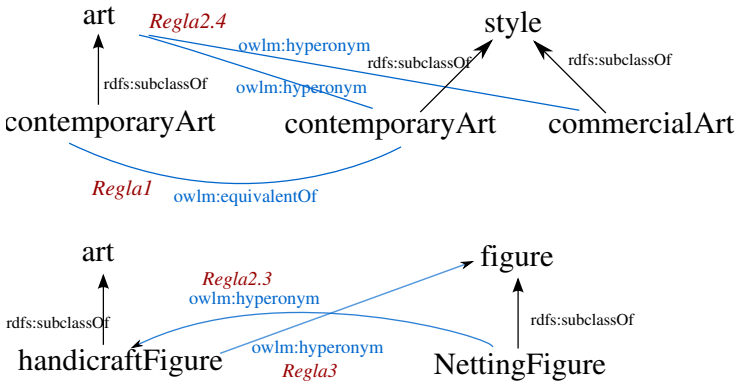


Figura 7.1: Ejemplo de alineamientos entre palabras compuestas

Las acepciones establecen la medida por excelencia pero la siguiente medida es relativamente más compleja por involucrar relaciones de similitud más complejas. Gracias al proceso de desambiguación, poseemos un buen número de correspondencias entre palabras por cada ontología. Al unificar las tablas de ambas ontologías se puede detectar solapamientos entre palabras coincidentes por nombre y consecuentemente, se puede identificar el tipo de correspondencia semántica que tienen con la clase. Una vez identificado el tipo de relación con sus clases, se puede propagar esta información a través de una heurística lógica establecida mediante los constructores OWL-M. No siempre se puede establecer una correspondencia entre los términos debido a la complejidad y al entramado posible de relaciones pero en algunos casos sí. El grupo de casos identificados son los siguientes:

- *Caso 1.* Cuando dos clases comparten el mismo nombre pero no la misma acepción. En este caso hemos supuesto que ambas clases son sinónimos. Es una suposición establecida para obtener resultados coherentes en la evaluación OAEI.
- *Caso 2.* Cuando dos clases comparten un sinónimo común pero no tienen el mismo nombre, entonces ambas clases son sinónimas.
- *Caso 3.* Cuando dos clases comparten un término común entonces tienen una acción, un verbo. En la propiedad no se identifica el verbo por cuestiones de simplicidad. Para realizar esto se debería de utilizar otros recursos y relacionar términos con acciones. Es una investigación ajena al proyecto.

El siguiente análisis está representado en el siguiente pseudo-código 7.2.

Por motivos de sencillez, especificamos los alineamientos por las reglas y casos posibles. En cada alineamiento figuran los conceptos implicados y la relación de unión entre ambos. En la comparativa de acepciones simplemente se usa el identificador de WordNet y ningún otro tipo de información. OWL-M ofrece la posibilidad de realizar relaciones más complejas entre clases pero por motivos de información disponible (recursos externos) y algoritmia (simples reglas y casos) no hemos establecido ningún tipo de camino compuesto.

## 7.3 Conclusión

Cabría esperar un capítulo más extenso y con una algoritmia más compleja pero como se ha defendido a lo largo de esta investigación el proceso de descubrimiento de acepciones incluye un gran número de fases que facilitan el mapeado de ontologías. El desarrollo de OMoCC ha sido costoso en tiempo y costoso en la depuración de la algoritmia derivada de la presencia continua de casos de recursividad en la sintaxis de OWL y en la recursividad planteada en el proceso.

Los alineamientos basados en equivalencias se determinan por comparación de acepciones entre términos de ambas ontologías. El resto de alineamientos

Código 7.2: **Pseudo-código: alineamientos de clases equivalentes**

```
//A partir de las clases de un modelo
ListClasses<URI> idConcept = ...

for (URI key : idConcept){

 //Cargamos el conceto vinculado a esa clase
 Concept c1 = (Concept) tableModel1.getElement(key, Concept.class);

 //Si la palabra está bien escrita se procede...
 if (!c1.isMisspelling()){
   //Si la palabra no es compuesta
  if (!c1.isCompoundWord()){
   //capturamos la anotación del identificador de la acepción en WordNet
   idWN = c1.getIDSense();
   //Buscamos ese mismo identificador en los conceptos del otro modelo.
     //Si existe el mismo identificador entonces
     //creamos un alineamiento al producir una correspondencia
     Alignment ali = new Alignment(c1,c2,OWLM.equivalentClass);

   //Es Caso1?
        Alignment ali = new Alignment(c1,c2,OWLM.synonymOf);
   //Es Caso2?

   //Es Caso3?

  }else{
    //cuando la palabra es compuesta
     //regla 1
     //reglas 2: 2.1 .. 2.4
     //regla 3
  }
 }
}
//Al finalizar los alineamientos son escritos en una nueva ontología
//fijando las correspondientes importaciones y prefijos
```

depende de las relaciones semánticas que se pueden establecer entre clases relacionadas por la información obtenida en el proceso de desambiguación. El tipo de relaciones en los alineamientos depende de los constructores definidos en OWL-M. Los resultados son almacenados en otro modelo donde se importan las ontologías originales. En definitiva, estos hechos conforman una nueva ontología totalmente compatible con herramientas, aplicaciones y otras ontologías para su uso según la naturaleza de cada una. Ejemplos del proceso de mapeado son explicados en el siguiente capítulo de evaluación.

El proceso es secuencial y se basa en determinar ciertos casos específicos acontecidos en la aparición de palabras compuestas y el posible repertorio de relaciones semánticas en las correspondencias de las tablas de almacenamiento de palabras. No es por tanto un proceso iterativo de refinamiento o matricial que requiera de un número indeterminado de pasos para establecer unos resultados. Los resultados son fijados sin supervisión humana y son coherentes a la interpretación de los diseñadores de los respectivos modelos.

# 8

# Evaluación

El trabajo de este capítulo contiene la evaluación de la calidad del alineamiento producido con OMoCC. Para comprender la evaluación de las ideas desarrolladas en OMoCC expondremos una serie de ejemplos. Conjuntamente introduciremos la evaluación del descubrimiento de la acepción para finalizar con la exposición de la evaluación de los alineamientos en el proceso de mapeado. Este hecho se debe a que los alineamientos del algoritmo dependen de la calidad de los resultados del descubrimiento de la acepción.

## 8.1 Evaluación del descubrimiento de las acepciones

Idealmente ambas evaluaciones, tanto del descubrimiento como del mapeado, deben de realizarse bajo los limites máximos definidos en un conjunto de suposiciones supervisadas. Es decir, la forma más correcta de evaluar es contrastar los resultados con anotaciones puestas por expertos bajo los dominios de la representación [131].

En el caso que nos concierne, *Senseval*[1] es considerado como el principal foro, conjunto de benchmarks y congresos para la evaluación en la desambiguación de palabras en corpus. *Senseval* define un conjunto de pruebas. Desde la primera hasta la cuarta, *Senseval* ha ido incrementado el ámbito incluyendo anotaciones semánticas y formas lógicas. *Senseval* tiene una rama de evaluación dedicada al conjunto de evaluaciones donde predominan las anotaciones semánticas, llamada *Semeval*.

En nuestro caso no podemos aplicar el conjunto de benchmarks de *Senseval* por enfocarse a estructuras de corpus lingüísticos donde las ontologías no están incluidas. Para proveer una solución a esta laguna de pruebas optamos por definir manualmente las acepciones de un conjunto de ontologías. Utilizamos las 16 ontologías de OAEI definidas dentro de las pruebas de *conference*. En cada ontología anotamos mediante OWL-M un identificador único de acepción

[1] http://www.senseval.org/

tanto de WordNet como de *Roget's thesaurus*, anotando manualmente un total de 870 clases. Al ser un dominio restringido al "mundo de las conferencias" no ha sido necesario realizar encuestas o discusiones sobre la viabilidad de asignar acepciones. Nos hemos limitado a utilizar las acepciones o palabras claves de WordNet y *Roget's thesaurus*. Al estar el contexto claro, la acepción de *chair* no recae sobre el significado de una silla sino sobre la persona quien preside un evento. Evidentemente el algoritmo ha de ser capaz de buscar la acepción acorde al dominio.

En el apéndice E introdujimos la evaluación por el número de aciertos usando clases SPC. Este análisis se puede resumir con los resultados de las 12 ontologías de la figura 8.1. De las 16 ontologías 4 se descartaron por problemas con el razonador al no dar soporte a las nuevos constructores. En la figura vemos una comparativa ordenada según el porcentaje de aciertos de cada ontología. El porcentaje de aciertos en las acepciones se mantiene alrededor del 50 %, exactamente 48.5 % con 7.12 de desviación.

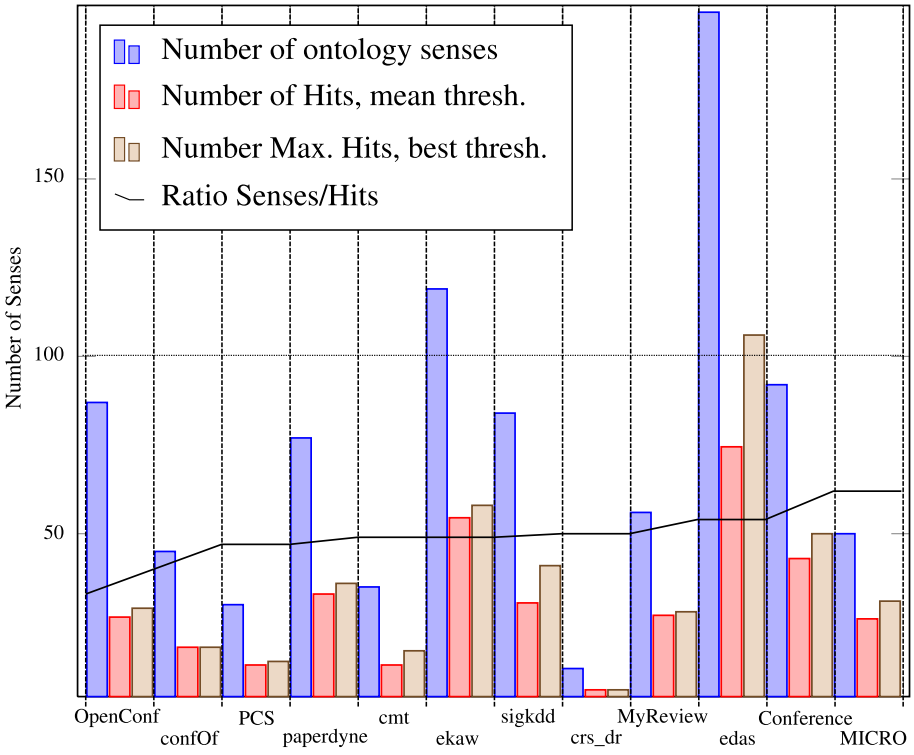


Figura 8.1: Comparativa ordenada respecto a los aciertos ponderados de cada ontología

Por la inexistencia de propuestas y de datos de referenciación previos no hay posibilidad de comparar los resultados con otras iniciativas provenientes de *Senseval* o de OAEI.

De la misma manera, destacar la importancia que desempeña WordNet en la correcta definición de acepciones y niveles jerárquicos tanto de la expresividad, categorización de ideas y contextualización de palabras con sus correspondientes términos. Muchos autores, y nosotros mismos dentro de nuestro conocimiento, vemos las deficiencias que adolece WordNet como herramienta para la desambiguación. WordNet [108] no fue diseñado para tareas de desambiguación y al ser la mejor opción disponible se convirtió en estándar de campañas de evaluación como *Senseval* y *Semeval*. Las aplicaciones de desambiguación sufren extremadamente las distinciones de matices que caracterizan a una palabra en WordNet [1], es decir, el nivel de granularidad en la definición, y en otros casos la escasez de palabras que complementan una acepción en su contexto [107]. En [1] múltiples soluciones son dadas: la creación de alternativos tesauros basados en WordNet y a un dominio en concreto, el incremento de información preferencial y de frecuencia, simplificación de jerarquías, acepciones mutuamente exclusivas, eliminación de enlaces incorrectos, entre otras propuestas. En nuestro caso, el haber dispuesto de información lógica (de exclusividad) entre acepciones al menos nos hubiera permitido determinar con mayor holgura el número correcto de acepciones y consecuentemente, haber visto incrementado el número de aciertos factibles por contexto. Un ejemplo, la palabra `conference` contiene tres acepciones [2] siendo factibles para este dominio tanto la primera como la última:

1. *a prearranged meeting for consulting or exchange of information or discussion (especially one with a formal agenda)*
2. *an association of sports teams that organizes matches for its members*
3. *a discussion among participants who have an agreed (serious) topic*

## 8.2 Evaluación de OMoCC

Hemos separado la evaluación de OMoCC en dos partes. La primera muestra los resultados en función del lenguaje de la representación sin la capacidad de evaluar pero pudiendo comprobar la utilidad de OWL-M. En la segunda parte, hemos adaptado la representación de los resultados para compararnos con propuestas de OAEI.

### 8.2.1 Representación de resultados

Desde nuestros conocimientos resulta interesante y comprensible los resultados que ofrece OWL-M. Primero introducimos un caso sencillo para facilitar la compresión de uno más complejo.

[2] http://wordnetweb.princeton.edu/perl/webwn?s=conference

Hemos representado un conjunto pequeño de conceptos, a nivel jerárquico, basados en el dominio de una biblioteca. El código de ambas ontologías está en el apéndice G. En la figura 8.2 están representadas las jerarquías de clases de ambas ontologías, siguiendo la apariencia del editor Protégé. Sus respectivos códigos están en el apéndice G. A la izquierda está la taxonomía de `OntoAuthor` y a la derecha, `OntoWriter`. El mapeado de ambas ontologías mediante OMoCC y la representación de los alineamientos con OWL-M están en color rojo. Estos son los enlaces de unión y la aparición de dos nuevos conceptos por el acontecimiento de un patrón jerárquico: *product* y *publication* (ver sección 5.3.1). El código está en el apéndice G. El algoritmo establece las siguientes relaciones entre conceptos, para no confundir con anotación extra de izquierda a derecha corresponden a sus respectivas ontologías acorde a la figura:

- La clase *Author* es `HypernymOf` *Writer*. A causa de la taxonomía en WordNet y por la común línea de acepciones se descubre que escritor es subclase de autor.
- La clase *Collection* es equivalente a *Collection* por compartir la misma acepción.
- La clase *Book* es `similarOf` a *Book*, por compartir un nombre común pero tener acepciones diferentes[3].
- La nueva clase *product* es subclase de *Creation* y superclase de *Book*, hecho debido a la detección de este concepto entre ambas acepciones de *Creation* y *Book* en WordNet.
- La nueva clase *publication*, le ocurre lo mismo pero entre *Work* y *Book*
- El *product* es una `actionRule` de *Work*, es un `Caso 3` (sección 7.2) una regla de acción sin conocimiento del verbo, es decir, una causa.
- La clase *CollectionBook* es una subclase de *Book*.

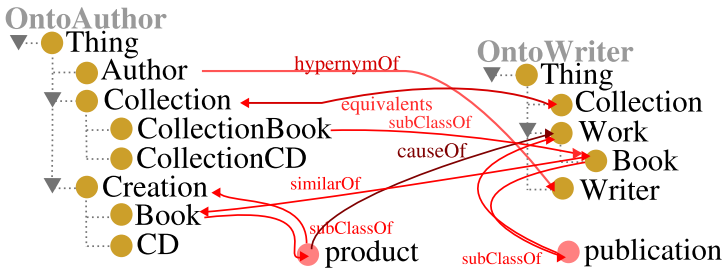


Figura 8.2: Mapeado entre ontologías mediante OWL-M

La conversión de OWL-M a *Alignment API* ha sido realizada teniendo en cuenta que las relaciones de equivalencia y similitud son relaciones de equiva-

[3] http://wordnetweb.princeton.edu/perl/webwn?s=book

lencia en *Alignment API* con la máxima similitud. El resto de constructores no son considerados por no existir una forma lógica de representarlos con exactitud. El código está en el apéndice G y en la figura 8.3 se aprecian los alineamientos en función de la utilización de este formato.

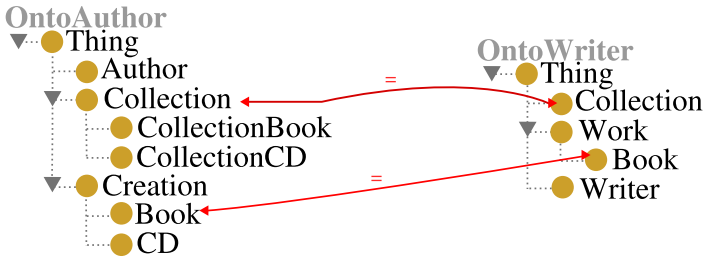


Figura 8.3: Mapeado entre ontologías mediante *Alignment API*

Con dos ontologías diferentes hemos procedido de la misma manera cotejando los resultados mediante ambas representaciones. Las ontologías son edas[4] y ekaw[5]. Por el gran volumen de relaciones procedemos a comentar algunas relaciones descubiertas y representadas mediante OWL-M (en apéndice G) y, su simplificación, con *Alignment API* (en apéndice G).

Destacamos algunos resultados:

Se detectan varias relaciones de acciones con la clase *Person* (código 8.1).

Código 8.1: **Las relaciones con la clase Person**

```
<rdf:Description rdf:about="http://edas#Person">
  <owlm:actionRule rdf:resource="http://ekaw#Organisation"/>
  <owlm:actionRule rdf:resource="http://swap.uib.es/2009/08/owlm#enrollee
      "/>
  <owl:equivalentClass rdf:resource="http://ekaw#Person"/>
  <owlm:actionRule rdf:resource="http://ekaw#Social_Event"/>
  <owlm:actionRule rdf:resource="http://ekaw#Presenter"/>
</rdf:Description>
```

Se crean nuevos conceptos entre jerarquías y WordNet o las ya existentes (código 8.2).

Código 8.2: **Algunos conceptos nuevos**

```
<rdf:Description rdf:about="http://ekaw#Individual_Presentation">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#presentation
      "/>
</rdf:Description>

<rdf:Description rdf:about="http://ekaw#Student">
  <rdfs:subClassOf rdf:resource="http://ekaw#enrollee"/>
</rdf:Description>
```

[4] http://oaei.ontologymatching.org/2010/conference/data/edas.owl
[5] http://oaei.ontologymatching.org/2010/conference/data/ekaw.owl

Se redefinen jerarquías donde participan palabras compuestas (código 8.3).

Código 8.3: **Nueva reestructuración**

```
<rdf:Description rdf:about="http://ekaw#Neutral_Review">
  <rdfs:subClassOf rdf:resource="http://edas#Review"/>
</rdf:Description>
```

Se detectan clases equivalentes, similares y sinónimos (código 8.4).

Código 8.4: **Clases similares y sinónimos**

```
<rdf:Description rdf:about="http://edas#RejectedPaper">
  <owlm:synonymOf rdf:resource="http://ekaw#Rejected_Paper"/>
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Presenter">
  <owlm:similarOf rdf:resource="http://ekaw#Presenter"/>
  <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Workshop">
  <owl:equivalentClass rdf:resource="http://ekaw#Workshop"/>
</rdf:Description>
```

Las relaciones identificadas mediante *Alignment API* se reducen a la detección de equivalencia entre términos con misma notación, como son: *Review, Person, Paper, Document, Workshop, Presenter* y *Conference*.

### 8.2.2 Evaluación en la plataforma SEALS

Como presentamos en la sección 3.4.1, tres medidas marcan la calidad de un algoritmo de mapeado: la precisión, *recall* y la *F-measure*, esta última corresponde a la media armónica de las dos primeras. La precisión corresponde al grupo de elementos que son clasificados correctamente. *Recall* es el porcentaje de elementos clasificados sobre el total.

La medición de estos tres parámetros de OMoCC se realiza en comparación con los resultados de la campaña de 2010 de OAEI[6]. Esta campaña se evaluó mediante una aplicación desplegada bajo el proyecto SEALS. Por motivos de disponibilidad de esta plataforma[7], la evaluación se ha realizado mediante la nueva plataforma del año 2011[8]. A pesar de este cambio, el conjunto de datos utilizado sigue siendo los mismos que en 2010. De la misma manera, la comparación con otras propuestas se ha basado en 2010 por la no disponibilidad de los resultados del 2011.

Los resultados para el grupo de ontologías de *conference* se muestran en la tabla 8.1. Cuando en el proceso participa la ontología `iasted` anotada con sus correspondientes acepciones, el razonador (Pellet[9] versión: 2.2.2) imposibilita el análisis por motivos de falta de memoria. Por eso, no se ha podido determinar ningún resultado en estas pruebas cuando participa dicha ontología.

[6] http://oaei.ontologymatching.org/2010/results/conference/index.html
[7] http://oaei.ontologymatching.org/2010/seals-eval.html
[8] http://oaei.ontologymatching.org/2011/seals-eval.html
[9] http://clarkparsia.com/pellet/

En líneas generales el valor de precisión es alto por ofrecer un número de resultados bastante acotado y certero con el real. Por este mismo motivo, el valor de *recall* es relativamente bajo ya que no se ofrecen todos los resultados esperados. Otro valor que influye considerablemente en este hecho es la contemplación de las propiedades. En OMoCC no se ha implementado la comparación de propiedades por su poco aporte a las hipótesis pero se enmarca en el trabajo futuro.

| **Ontologies** | **Precision** | **Recall** | **F-measure** |
|---|---|---|---|
| cmt-confOf | 0.8 | 0.25 | 0.381 |
| cmt-conference | 0.667 | 0.25 | 0.364 |
| cmt-edas | 1 | 0.615 | 0.762 |
| cmt-ekaw | 1 | 0.456 | 0.625 |
| cmt-iasted | 0 | 0 | 0 |
| cmt-sigkdd | 1 | 0.5 | 0.667 |
| confOf-edas | 0.8 | 0.421 | 0.552 |
| confOf-ekaw | 0.889 | 0.4 | 0.552 |
| confOf-iasted | 0 | 0 | 0 |
| confOf-sigkdd | 0.8 | 0.571 | 0.667 |
| conference-confOf | 0.7 | 0.467 | 0.56 |
| conference-edas | 0.636 | 0.412 | 0.5 |
| conference-ekaw | 0.692 | 0.36 | 0.474 |
| conference-iasted | 0 | 0 | 0 |
| conference-sigkdd | 0.8 | 0.533 | 0.64 |
| edas-ekaw | 0.714 | 0.217 | 0.333 |
| edas-iasted | 0 | 0 | 0 |
| edas-sigkdd | 0.875 | 0.467 | 0.933 |
| ekaw-iasted | 0 | 0 | 0 |
| ekaw-sigkdd | 1 | 0.636 | 0.778 |
| iasted-sigkdd | 0 | 0 | 0 |

Cuadro 8.1: Resultados para las ontologías de *conference*

Para mostrar en detalle los resultados hemos elegido el par *cmt* y *conference*. En OAEI, los alineamientos marcados como referencia están formados por las siguientes 14 equivalencias entre clases: *Conference = Conference_volume, Preference = Review_preference, Author = Regular_author, Person = Person, Co-author = Contribution_co-author, PaperAbstract = Abstract, Document = Conference_document, Review = Review, Conference = Conference, Program-Committee = Program_committee, Chairman = Chair, SubjectArea = Topic* y 3 equivalencias entre propiedades: *assignedByReviewer = invited_by, email = has_an_email y assignExternalReviewer = invites_co-reviewers.*

OMoCC usando la representación de AlignmentAPI sólo devuelve 5 equivalencias entre el par de clases con: *Paper, Person, Conference, Review, y Reviewer*. OMoCC usando OWL-M considera otro tipo de relaciones donde

las clases anteriores involucradas en los resultados son: *Regular_author* es una subclase de *Author, Review_preference* es una subclase de *preference*, etc. *ProgramCommitee* es un sinónimo de *Program_committee*, *Chairman* es detectado como un error, y *Topic* y *SubjectArea* no son identificados como nada.

En el segundo caso, con OMoCC usando OWL-M, si ignoramos las diferentes relaciones semánticas entre términos y las acepciones solamente 4 de los 17 alineamientos no serian detectados. De estos 4, 3 de ellos son alineamientos de propiedades. En este caso, OMoCC obtendría un 82 % de *recall*, la precisión sería del 100 %. Obviamente, esta precisión y *recall* está considerada bajo los 17 casos considerados por OAEI, y no los 52 resultados que OMoCC proporciona con OWL-M.

La precisión menor se obtiene con el par *conference* y *edas*. La mayor precisión en varios casos: *cmt-edas, cmt-ekaw, cmt-sigkdd* y *ekaw-iasted*. Se aprecia la participación de la ontología *conference* en aquellos resultados con peores bondades. Uno de los posibles motivos que hemos identificado es el porcentaje de palabras compuestas, un 84 %. La ontología *conference* es la segunda ontología con más palabras compuestas. Encabeza la lista, la ontología *edas* con un 85 %. La diferencia entre ambas está en la frecuencia de términos idénticos de la primera, donde palabras como: *conference, call, contribution* y *author* entre otras, aparecen en la mayoría de estas palabras compuestas. Es decir, su frecuencia de aparición en palabras compuestas es elevada. Por esta causa, al ser un grupo reducido de palabras, no se producen correspondencias con otras palabras de recursos externos. Esto permitiría descubrir relaciones entre ambas estructuras que a la postre facilitarían establecimientos de correspondencias y de patrones.

Si comparamos los resultados de OMoCC con respecto a las propuestas de 2010, vemos tanto en la tabla 8.2 y visualmente en la figura 8.4 su posición relativa. OMoCC al igual que CODI no requiere del establecimiento de un grado de umbral para suponer algún tipo de relación entre conceptos. Por el contrario, en cada propuesta se ha fijado el mejor umbral para obtener el mejor rendimiento. Visualmente, la precisión de OMoCC es elevada pero no destaca significativamente en *recall* por las causas que hemos podido observar con la explicación del ejemplo *cmt-conference*.

Realizamos el test sobre la evaluación específica del repertorio de casos de la librería[10]. El objetivo de este benchmark es proveer de un conjunto de casos que van incrementando o decrementando de manera progresiva y estable funcionalidades de la representación. Esta evaluación consta de 89 casos de estudio, 89 pares de ontologías. Los resultados son representados en la figura 8.5. El primer caso, el 101, es la ontología de referencia, donde OMoCC obtiene 0.888 en precisión y 0.329 en *recall*. De media en todas las pruebas, OMoCC obtiene 0.89 y 0.49 en precisión y *recall* respectivamente. Como se puede apreciar en la figura hay una serie de casos en los que no se proporciona ningún resultado. Las causas son las siguientes: en 202 algunos

[10] http://oaei.ontologymatching.org/2011/benchmarks/

| Matcher | Confidence threshold | P. | R. | F. |
|---|---|---|---|---|
| AgrM | 0.66 | 0.53 | 0.62 | 0.58 |
| AROMA | 0.49 | 0.36 | 0.49 | 0.42 |
| ASMOV | 0.22 | 0.57 | 0.63 | 0.6 |
| CODI | * | 0.86 | 0.48 | 0.62 |
| Eff2Match | 0.84 | 0.61 | 0.58 | 0.6 |
| Falcon | 0.87 | 0.74 | 0.49 | 0.59 |
| GeRMeSMB | 0.87 | 0.37 | 0.51 | 0.43 |
| SOBOM | 0.35 | 0.56 | 0.56 | 0.56 |
| **OMoCC** | * | 0.74 | 0.53 | 0.56 |

Cuadro 8.2: Comparativa entre propuestas de 2010

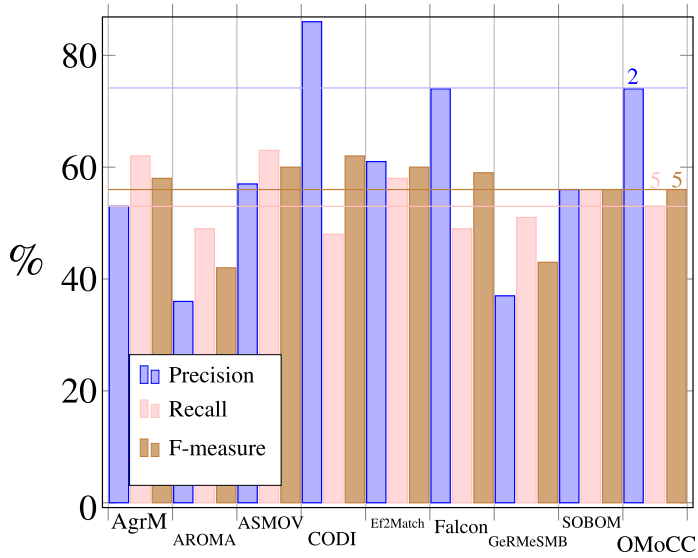


Figura 8.4: Comparativa entre propuestas de 2010

nombres están reemplazados aleatoriamente; en 237, 258-$i$ por no existir una jerarquía; y en 249, 253-$i$ y 259-$i$ la nomenclatura de las clases es aleatoria (no son términos disponibles en un diccionario).

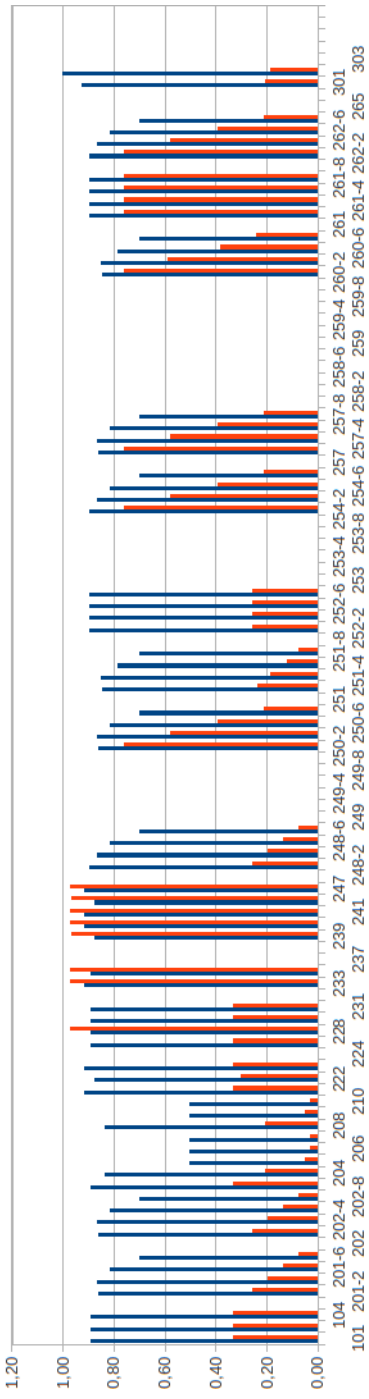


Figura 8.5: Resultados de OMoCC sobre la prueba *benchmark*. En azul la precisión y en rojo *recall*

El test de OAEI respecto a *anatomy* [11] no ha obtenido resultados. Se compone de un par de ontologias *Mouse* y *Human*. La ontología *Mouse* contiene clases nombradas siguiendo este patrón alfanumérico: "MA_"+ 7 dígitos. Los valores no tienen ningún significado aparente solo coinciden por MA. De manera similar, la ontología *Human* contiene las clases nombradas mediante: "NCI_C"+5 dígitos. Los dígitos no aportan nada al significado y los términos "MA" y "NCI_C" no coinciden por lo que OMoCC no ofrece ningún resultado.

## 8.3 Conclusión

Los resultados son variables ya que dependen de las decisiones que tome el diseñador durante la modelización del dominio (jerarquía, selección de términos, palabras compuestas, nombrado de propiedades, individuales, etc.) y por características de WordNet en términos de granularidad y existencia de palabras. Por el planteamiento del algoritmo, la evaluación de las acepciones no se ha podido realizar mediante una comparación con otras propuestas. Esto ha propiciado la creación de un corpus de ontologías que contiene las acepciones anotadas con los identificadores de WordNet y de *Roget's thesaurus*. En base a este corpus la calidad de las acepciones ronda el 50 % de aciertos.

Los alineamientos se han evaluado con las técnicas y corpus definidos en OAEI en la campaña 2010. OAEI coteja los resultados que estén representados en el formato de *Alignment API*. Por tal hecho, hemos tenido que convertir y reducir el número de resultados obtenidos con OWL-M. Esta reducción, analizada en detalle con varios casos, demuestra el notable volumen de resultados obtenidos con OMoCC en función de cada una de las representaciones. De la misma manera, no hemos podido evaluar el total de los resultados representados en OWL-M por la complejidad de tal tarea. Por el contrario, si hemos evaluado los resultados reducidos. Estos resultados demuestran que la precisión de OMoCC es bastante elevada estando entre las dos primeras propuestas de 2010 y en términos de *recall* entre las 5 primeras propuestas de ese mismo año.

Habría que evaluar el mapeado en función de dominios con diferentes grados de solapamiento en términos contextuales. Este conjunto de pruebas con dominios dispares a diferentes niveles sí que permitiría ver realmente el funcionamiento de OMoCC y de otros algoritmos bajo dominios diferentes.

[11] http://web.informatik.uni-mannheim.de/oaei/anatomy11/index.html

# Part III

# Conclusions

# 9

# Conclusions

This final chapter summarises statements, ideas, algorithms and experiments of this thesis. It includes the main contributions the author has identified and it contains a discussion of results and future work.

## 9.1 Thesis summary

In this document, we have addressed design of mapping algorithms to evidence that the minimal measure of comparison is the meaning and the necessity of represent the results in the same format than original data sources. In chapter 2, the reader has a global and plain vision of the disciplines regarding with the ontology mapping field. From the point of view of multiples philosophers and relevant authors regarding with this topic (Artificial Intelligence, Information Retrieval, Natural Language Processing, Data Mining, Text Mining, Word Sense Disambiguation, among others), we established the fundamental pillars of our baseline. At that point, we achieved the enough background to introduce the rest of aspects, phases, and related work in chapter 3. In order to explain Ontology Mapping (OM) phases, it follows the list of phases derived from Data Mining techniques that overlap with OM phases in a perfect way: linguistic preprocessing, similarity metrics: lexical, structures and in function of external resources: dictionaries, thesaurus, upper-ontologies or previous alignments, and evaluation metrics and campaigns. Keeping these explanations, we commented 20 works from 2002 to present. This chapter ends with an interesting discussion (in section 3.7) of the several works according some features. Those features regard information used, real mapped information, lexical, structural and combinational measures, evaluation method, language of representation of alignments, and finally, the meaning or context data that authors used to decide the alignment. 4 of 24 proposals take into account some contextual aspects of the representation and only one approach uses the meaning as part of the OM process.

Next part of document is split in three issues which define this work: meaning discovery of ontology elements (in chapter 5), results representation (in chapter 6) and the combination of both theories in the development of OMoCC (in chapter 7). Analysing metrics to deduce the meaning, we defined our conception of the *context* and with that definition, we discovered that certain elements appear more than others and are more relevant to define the context and the meaning of the rest the elements due to structural characteristics and other specific constructors. We called them structural predominant classes (SPC). Using only these classes we reduced the amount of resources for considering both metrics direct patterns and correspondences between ontology and WordNet. To maximize the use of alignments in final applications, we have to define alignments via semantic representations. Thus, OWL-M definition contains an extension of OWL based constructors addressed to alignments functionalities. All of them follow logic and formal nomenclature defined in OWL by W3C. Last chapter contains high-level explanation of our OMoCC algorithm. Algorithm evaluation 8 compares our results with another proposes presented at OAEI initiative in 2010. We have to simplify alignment representation to adapt OWL-M constructors at Alignment API, losing expressiveness, alignments and semantic information. In terms of disambiguation, we cannot evaluate our disambiguation algorithm with general benchmarks. However, we have used some ontologies of OAEI campaigns to evaluate our results.

## 9.2 Contributions

We list the main contributions of this thesis which address and extend the initial objectives in section 1.2:

1. *A systematic survey of Ontology Matching.* Chapter 3 contains an extensive analysis of main approaches and interesting discussion of each one proposal, from 2004 to 2010, in terms of meaning and semantic metrics. That point of view, semantic metric was not used and it should take part as fundamental metric.
2. *A realistic approximation of mapping information in function of meaning.* From introduction to chapter 5, we explain basic ideas regarding with importance of meaning to establish effective alignments in philosophic, logic, and linguistic terms.
3. *A manual annotated corpus of ontology classes for disambiguation techniques in WSD disciplines.* Evaluation of algorithms in disambiguation ontology classes was not able to be compared with other initiatives since there was not a reference corpus. We have defined a corpus set with 16 ontologies with 1422 annotated words with two identifiers at WordNet and Roget's thesaurus.
4. *A novel definition of a synthesis of elements of an ontology.* We have determined a set of metrics to synthesise elements, classes, considering

contextual information. These classes were called Structural Predominant Classes (SPC), and they can be useful for cache tasks, indexing information, compress information, etc.

5. *An extended vocabulary for alignment descriptions and descriptive annotations of tasks in the mapping process.* That set of constructors is called OWL-M. Basically, they are based on syntax and logic of OWL and can be reused along other layers in the application architecture.

Other contributions are the next and they can be downloaded from[1].

1. *An OWL-GEXF parser.* In order to visualize and make easier some visual calculation (i.e. distances, clustering, network analysis, etc.) we have developed a parser between OWL and GEXF (Graph Exchange XML Format). It is a language to describe complex networks structures, their associated data and dynamics. It is fully supported with Gephi project[2]. Gephi is an interactive visualization and exploration platform for all kind of networks and complex systems. The figure 5.4 is created using this tool.
2. *An Cytoscape plug-in to load OWL ontologies.* In the same way, we have developed a plug-in for Cytoscape to visualize ontologies. Cytoscape[3] is an open source platform for complex network analysis and visualization. Both developments serve to analysis some assumptions that were tested experimentally.

## 9.3 Summary of challenges achieved

In "Ten Challenges for Ontology Matching" [141] which contains an analysis of the main trends that an OM algorithm should be dealt, the challenges discussed are: large-scale evaluation, performance of ontology-matching techniques, discovering missing background knowledge, uncertainty in ontology matching, matcher selection and self-configuration, user involvement, explanation of matching results, social and collaborative ontology matching, alignment management: infrastructure and support, and reasoning with alignments. Therefore, this thesis according with the authors suggestions is addressed to these issues:

1. *Large-scale evaluation.* In the different issues about this topics there are: *"OAEI campaigns and scalability characteristics, the need for more accurate evaluation quality measures, the need for evaluation methods grounded on a deep analysis of the matching problem space, efforts on meta-matching systems, composing matchers and on Alignment API".* Current representing languages are Alignment API and, more recently,

[1] https://sourceforge.net/projects/owlgexfparser/
[2] http://gephi.org/
[3] http://www.cytoscape.org/

EDOAL. EDOAL has the capability to combine classes and properties setting complex constructors. However, EDOAL is based on Alignment API does not represent alignments with semantic information. It keeps only one relationship (=) and degree value. On the contrary, OWL-M offers a set of constructors with full compatibility with OWL, it gives the possibility to represent a deep analysis of alignment space and OWL-M annotations contains meta information about detailed aspects of algorithm and metric.

2. *Performance of ontology-matching techniques. "The fact that some systems run out of memory on some test cases, although being fast on the other test cases, suggests that their performance time is achieved by using a large amount of main memory. Therefore, usage of main memory should also be taken into account"*. Structural predominant classes synthesis 60% ontology elements when we consider contextual information. Furthermore, typical matrix comparison, where algorithms compare all data, is not efficient than our patrons and correspondence rules.
3. *Discovering missing background knowledge. "One of the sources of difficulty for the matching tasks is that ontologies are designed with certain background knowledge and in a certain context, which unfortunately do not become part of the ontology specification, and, thus, are not available to matchers. Hence, the lack of background knowledge increases the difficulty of the matching task*". In that point, we address our hypothesis to exploit that implicit data using discovering contextual or meaning techniques of third related areas and, of course, using reasoning tools to manage ontologies.
4. *Uncertainty in ontology matching. "Often the best way to resolve uncertainty in matching process is using a similarity matrix among all ontology elements, and in several iterations algorithms rule out some alignments"*. Uncertainty is redefined by comparing $k$ alignments (metrics) in each iteration. In general, uncertainty is dealt through a numeric value, that value does not represent the multiples suppositions of each metric. We address our model to represent that uncertainty with semantic correspondence, and finally, a process has to take a decision about the relative importance of each semantic correspondence and it chooses one (or a combination of them, a OWL-M path).
5. *Matcher selection and self-configuration. "There are many matchers that are available nowadays. Often these perform well in some cases and not so well in some other cases"*. We like to introduce another fact, ontology matching algorithm should be transparent for applications and users. Algorithm should be able to achieve efficient self-configuration. In this sense, we designed OMoCC without the necessity of a "configurable threshold". Of course, one part of OMoCC uses an experimental threshold to analysis the external resource but if we want to guarantee efficient results we have to define boundaries. As when people take decisions, that are often performed in a limited time in both non- a critical moments and for that, we

define knowledge boundaries. Regarding with matcher selection, we only define a minimum metric: the meaning, the rest of matchers complement and help that first decision.

6. *Explanation of matching results. "In order for matching systems to gain a wider acceptance, it will be necessary that they can provide arguments for their results to users or to other programs that use them. In fact, alignments produced by matching systems may not be intuitively obvious to human users, and therefore, they need to be explained"*. For example, when an alignment returns a degree of 0.709 nobody knows exactly the meaning of that value in terms of semantic suppositions in a semantic representation. Clearly, OWL-M defines semantic constructors and annotations to describe the alignment and the process, respectively. OWL-M alignments have the same readability than OWL language.
7. *Reasoning with alignments. "The ultimate goal of matching ontologies is to use alignments. For this purpose, they should be attributed a semantics"*. Applications should be able to use alignments in the same way that they use original ontologies. For that reason, OWL-M, which is based on OWL, eases the integration of alignments in the rest of layers of the final application.

## 9.4 Future Work and applications

### 9.4.1 Extension of the current work

We have classified this extension in terms of development work: two aspects beyond our hypothesis but fundamental to complete our work and also two new ideas which will try to develop in further works.

We should complement this work through next two aspects:

1. *Mapping ontology properties.* In the development of OMoCC, we have only considered classes as main elements to be mapped. However, properties have a crucial role in integration tasks since they define relationships among classes and characteristics of the attributes. Classes derived using OMoCC inherits the original properties of the sources. In some cases, these classes are a logical combination of other domain and range and can be combined with OWL-M and other future constructors that we are not considered yet.
2. *Taking part in OAEI 2012.* We have used OAEI 2010 dataset and benchmark for OMoCC evaluation. Our adaptation of OMoCC results is based on equivalent properties but we will need work in better types of adaptations to participate in the next campaign of OAEI.

Indirectly, from the knowledge learned in this thesis, two other research projects are envisioned:

1. *Deep design and definition of a collaborative schema for disambiguation techniques*, i.e. Wikipedia/DBpedia. External resources in retrieval information tasks are the pillars of suitable results. However, availability of these resources is not enough mature in terms of semantic characteristics. Web evolution tends to share data instead of the creation of new data. The creation of new contents will come from a collaborative work and that task requires combination and integration of knowledge. For that reason, an external source well structured will be able to do integrate task keeping lines created which use the persons that managing the knowledge.
2. *Conflicts*. In this thesis, we have not dealt with conflict of classes. They happen when disjoint axioms are part of the definition and classes from other ontologies are supposed to be part of the "forbidden" hierarchic. But, when the meaning is take into account this type of conflicts are caused by an inadequate design schema. However, mapping properties with universal and existential quantifiers introduce a higher level of complexity. Dealing conflicts with properties is a non-deterministic polynomial time problem. Ram et *al.* [127] introduced this topic in semantic integration problems of databases in 2004.

### 9.4.2 Possible applications

The findings of this thesis can be exploited in the following applications. In general, when *heterogeneous data across multiples sources have to be combined*, typical of integration case. For instance, we like to introduce data problematic in SOCIB [4] (a Coastal Ocean Observing and Forecasting System located in the Balearic Islands). "SOCIB is a multi-platform distributed and integrated system that will provide streams of oceanographic data and modelling services to support operational oceanography in a European and international framework, therefore also contributing to the needs of marine and coastal research in a global change context". They combine a heterogeneous observing systems to collect data. The amount of knowledge available from a simple drifter, glider or simple beach camera monitoring contains thousands of simple measures that also appear in other monitoring systems. Nowadays, they include data from several tools in multiple data schemas in an isolated way. They use a complex logic layers to extract punctual knowledge. But, that logic has to be defined each time and also representation layers. Next step in SOCIB data model is to discover implicit relationships among current data sources and future monitoring and predictive systems. To solve this issue, it is necessary to use data integration techniques among different data sources. In this sense, OWL-base semantic models give full-support to future task of information extraction, data crossing, and data representation where OMoCC paradigm can solve data heterogeneous problems.

[4] http://www.socib.eu/

## 9.5 Final words

In 1999, Ram et *al.* said:"To achieve semantic interoperability, systems must be able to exchange data in such a way that the precise meaning of the data is readily accessible and the data can be translated by any system into a form that it understands" [128]. Some time has passed but the idea remains the same, with this work we have contributed to that goal. We have presented the baseline, a pair of hypothesis, a systematic analysis of related work, and we have defined a no novel metric, being the correct metric in this field and language layer, and evaluated the great development of the algorithm throughout creating a new ontology corpus and using the referential benchmark.

# Parte IV

# Appendixes

# A

# Propiedades con la partícula 'has' en su nombre

En este análisis se ha determinado el porcentaje de propiedades que contienen el término 'has' en las ontologías disponibles en la web. Con el objetivo de encontrar un amplio repertorio de ontologías se diseñó un *web crawler*. De los 3179 enlaces encontrados se descartaron aquellos inaccesibles y aquellas ontologías con ninguna clase. Por lo que se obtuvo un total de 724 ontologías.

Los resultados están en la tabla A.1. Las ontologías se han agrupado por el número de propiedades. La segunda columna contiene el número de ontologías con en ese rango de propiedades. La tercera columna representa el número medio de propiedades dentro de esa rango. La cuarta columna contiene el número medio de propiedades con la partícula 'has' y la última columna, el porcentaje de estas propiedades sobre el total.

| Propiedades | Ontologías | $\overline{propiedades}$ | Propiedades con `has` | % |
|---|---|---|---|---|
| [600-501] | 5 | 546.2 | 13.2 | 0.02 |
| [500-401] | 0 | 0 | 0 | 0 |
| [400-301] | 5 | 315 | 12.8 | 0.04 |
| [300-201] | 12 | 255.5 | 8.17 | 0.03 |
| [200-101] | 20 | 151.15 | 43.3 | 0.29 |
| [100-71] | 29 | 81.1 | 23.93 | 0.3 |
| [70-41] | 148 | 56.98 | 18.51 | 0.32 |
| [40-11] | 184 | 24.54 | 5.71 | 0.23 |
| [11-0] | 311 | 4.51 | 1.49 | 0.33 |
| | | | **Total:** | 0.23 |

Cuadro A.1: Porcentaje de propiedades con la partícula 'has'

El número medio de propiedades con esta partícula es del 23 %. Destacar que cuando estas ontologías poseen un menor número de propiedades éstas a su vez contienen un mayor porcentaje de propiedades con 'has'. Una de

las posibles causas es la creación manual de las mismas. A menor tamaño es más factible que se hayan creado manualmente y tengan un fin específico. Las ontologías con un gran volumen de propiedades suelen ser taxonomías o han sido creadas a partir de alguna base de datos, donde la selección del nombre se rige por un patrón alfanumérico, sin relación alguna con la interpretación de la relación entre las clases del dominio y del rango. Además, el conocido manual para la creación de ontologías del editor de Protégé hace un uso considerable de tal partícula.

# B

# Estudio del tipo de correspondencias

Se ha realizado un estudio para contabilizar el tipo de correspondencias entre palabras que se producen en diferentes ontologías. Este estudio es útil para computarlas. Las ontologías corresponden al dominio de conferencias definidas por OAEI de la campaña de 2010 [1].

Los resultados de este análisis se muestran en la tabla B.1. La primera columna contiene el nombre de la ontología. El resto de columnas representa mediante acrónimos el tipo de correspondencia: relaciones entre términos (TT), relaciones entre términos y palabras simples (TSW), relaciones entre términos y sinónimos (TSy), relaciones entre términos y propiedades (TNP), relaciones entre palabras simples y sinónimos (SWSY), relaciones entre palabras simples y propiedades (SWP), y relaciones entre sinónimos y propiedades (SYNP). Por tener una misma función las categorías siguientes de merónimos, holónimos, hiperónimos e hipónimos son considerados como *palabras simples* aplicando a cada una de ellas las restricciones oportunas comentadas anteriormente.

Por coincidencias de palabras, correspondencias útiles, hay un total de 593, un 92 % de ellas son correspondencias donde participan términos y un 56 % se deben a correspondencias entre términos. Es decir, la mayoría de las coincidencias de palabras se produce por la participación de algún término del *gloss*. En términos probabilísticos es compresible pues la mayoría de los palabras de WordNet están en el *gloss*.

[1] http://oaei.ontologymatching.org/2010/conference/index.html

| | **Tipo de relación** | | | | | | |
|---|---|---|---|---|---|---|---|
| **Ontología** | TT | TSW | TSy | TNP | SWSY | SWP | SYNP |
| cmt | 21 | 5 | 3 | 5 | 4 | 0 | 0 |
| conference | 119 | 29 | 31 | 11 | 5 | 0 | 6 |
| confious | 40 | 23 | 9 | 0 | 0 | 0 | 0 |
| confof | 53 | 14 | 4 | 3 | 0 | 0 | 0 |
| crs_dr | 13 | 10 | 1 | 0 | 2 | 0 | 0 |
| edas | 150 | 35 | 65 | 2 | 2 | 0 | 0 |
| ekaw | 23 | 8 | 19 | 0 | 0 | 0 | 0 |
| micro | 23 | 16 | 12 | 8 | 2 | 0 | 0 |
| myreview | 13 | 5 | 6 | 6 | 0 | 0 | 0 |
| openconf | 39 | 6 | 14 | 9 | 0 | 7 | 0 |
| paperdyne | 29 | 5 | 5 | 0 | 1 | 0 | 0 |
| pcs | 8 | 8 | 4 | 6 | 0 | 0 | 0 |
| sigkkd | 2 | 3 | 1 | 0 | 0 | 0 | 0 |
| **Total** | 533 | 167 | 174 | 50 | 16 | 7 | 6 |
| **TOTAL** | 953 | | | | | | |
| **%** | 0,56 | 0,18 | 0,18 | 0,05 | 0,02 | 0,01 | 0,01 |

Cuadro B.1: Tipos de correspondencias por ontología

# C

# Valoración de las correspondencias

En este anexo presentamos el análisis para ponderar adecuadamente las correspondencias. El peso de cada tipo de correspondencia determina el grado de vinculación de dos acepciones. Valorar adecuadamente las correspondencias es crucial para el correcto establecimiento de acepciones.

El criterio elegido para valorar cada uno de los pesos es contrastarlos con el número de aciertos en las acepciones. En este punto, el lector no habrá sido introducido al proceso del cálculo de las acepciones pues será explicado en futuras secciones (anexo E). Para simplificar la exposición de hechos ya comentados, huelga decir que se utilizan un total de 12 ontologías. Todas ellas presentan 593 clases y un total de 884 palabras o acepciones a descubrir. Se ha anotado manualmente cada una de las acepciones de las 884 palabras.

Como ya hemos mencionado, determinar aquellas correspondencias más significativas a la acepción, es básicamente, determinar el peso de la palabra según el diseño del recurso externo para ver cual es el aporte de su relación con la posible acepción. Los factores que afectan a esta problemática son el diseño de WordNet tanto en su estructura como en el uso de palabras, y la estructura y elección de palabras de una ontología.

Hemos planteado tres tipos de graduaciones en función de una serie de criterios. Los pesos de los tres criterios están representados en la tabla C.1. En la primera columna figura el tipo de correspondencia donde se aplica cada caso. No aparecen los casos de homónimos, hiperónimos, etc. porque hemos considerado que tienen el mismo peso que una palabra simple. La segunda columna contiene los pesos denominados como *simples*, son unos pesos constantes con valores 1 y 2. Se valora con mayor peso las relaciones donde participe un elemento que no sea un término. En la tercera columna, pesos por *frecuencia*, los valores corresponden a la frecuencia de aparición de cada una de las relaciones (ver la última fila de la tabla B.1). Se valora con mayor peso las relaciones donde aparecen términos. La última columna contiene valoraciones a la inversa de la frecuencia de aparición. Es decir, se valoran con mayor escala aquellas relaciones donde no aparezcan términos.

| Tipo correspondencia | Peso *Simple* | Peso *Frecuencia* | Peso *Semántico* |
|---|---|---|---|
| término ∼ término | 1 | 6 | 1 |
| término ∼ palabra simple | 2 | 2 | 3 |
| término ∼ sinónimo | 2 | 2 | 3 |
| término ∼ propiedad | 1 | 1 | 1 |
| palabra simple ∼ palabra simple | 2 | 2 | 5 |
| sinónimo ∼ palabra simple | 2 | 1 | 5 |
| sinónimo ∼ propiedad | 2 | 1 | 5 |
| palabra simple ∼ propiedad | 2 | 1 | 5 |
| resto casos | 1 | 2 | 1 |

Cuadro C.1: Tabla de valoraciones de las diferentes correspondencias, donde en cada relación se aplican las correspondientes restricciones, i.e. por lo general, en ambas partes el concepto ha de ser diferente

| Ontología | Clases | Acepciones | Aciertos por | | |
|---|---|---|---|---|---|
| | | | peso *simple* | peso *frecuencia* | peso *semántico* |
| confOf.owl | 44 | 45 | 18 | 18 | 18 |
| paperdyne.owl | 48 | 77 | 20 | 21 | 22 |
| sigkdd.owl | 52 | 84 | 30 | 36 | 28 |
| crs_dr.owl | 19 | 12 | 6 | 5 | 6 |
| ekaw.owl | 74 | 119 | 47 | 58 | 44 |
| edas.owl | 105 | 197 | 54 | 62 | 54 |
| PCS.owl | 25 | 30 | 9 | 12 | 9 |
| MyReview.owl | 39 | 56 | 22 | 25 | 21 |
| MICRO.owl | 36 | 50 | 33 | 28 | 32 |
| cmt.owl | 29 | 35 | 20 | 15 | 19 |
| OpenConf.owl | 62 | 87 | 22 | 19 | 23 |
| Conference.owl | 60 | 92 | 39 | 46 | 37 |
| | | | | | |
| Total | 593 | 884 | 320 | **345** | 313 |

Cuadro C.2: Resultados de los tres criterios para cada ontología

En la tabla C.2 se muestra el porcentaje de aciertos para las tres configuraciones. El total es la suma de todos los aciertos. El mejor número de aciertos se obtiene con la segunda configuración de valoraciones. Los aciertos se producen sobre las acepciones (*senses*) y no sobre las clases por la aparición de palabras compuestas. Sorprende obtener resultados más favorables con valoraciones basadas en la frecuencia de apariciones de correspondencias que con valoraciones sobre su inversa. Con la configuración por *frecuencias*, las palabras lejanas con coincidencias sobre términos en sus definiciones producen mejores resultados que palabras relacionadas por su estructura. Este hecho sólo depende del diseño de WordNet.

WordNet [108] no fue diseñado para actividades de la comunidad de desambiguación de términos y al ser la mejor opción disponible se convirtió en estándar en este tipo de tareas. Las aplicaciones de desambiguación sufren extremadamente las distinciones de matices que caracterizan a una palabra y en otros casos la escasez de palabras que complementan una acepción en su contexto [1; 9; 107].

# D

# Umbral de búsqueda en el recurso externo

En este anexo se justifica el valor máximo de búsqueda de elementos en el recurso externo para evitar el crecimiento continuo de recursos. Este nivel de poda se ha fijado experimentalmente a una longitud máxima de 4. El valor real es 8 para un nodo con relaciones de jerarquía (subclases y superclases) en la ontología.

Este valor se debe a una serie de consideraciones:

- Primera, la máxima longitud de relaciones jerárquicas en WordNet es de 19. En la tabla D.2 están representados todos los descendientes del concepto raíz[1] de WordNet. La mayoría de conceptos se agrupan en los niveles intermedios lo que posibilita que con una longitud de 4 se pueden alcanzar estos niveles propiciando un aumento de coincidencias.
- La segunda consideración se basa en la distancia conceptual de los conceptos en la jerarquía. Esta distancia no suele ser grande aunque depende del diseño de WordNet y del diseñador de la ontología. Por ejemplo, no es habitual jerarquías de este tipo en una ontología: `España` con `entidad abstracta` y `pajaro` con `entidad`. La cantidad de información entre los conceptos generales y específicos es elevada.
- Tercera consideración, se ha realizado un estudio empírico para determinar el nivel -la distancia entre ambos conceptos- donde se producen las coincidencias por patrones de jerarquía. Las relaciones de jerarquía son las más abundantes en WordNet. Este estudio esta representado en la tabla D.1. Cada fila corresponde a un nivel y las columnas son ontologías del grupo de conferencias de OAEI 2010. Por cada ontología y nivel, se muestra el total de patrones establecidos por el total de relaciones analizadas (penúltima fila).

[1] Vinculados a la primera acepción

| Nivel | ontología | | | | | | | | | | | | | TOTAL | % |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | cmt | cocus | conference | confOf | crs_dr | edas | ekaw | micro | myreview | openconf | paperdyne | pcs | sigkdd | | |
| **1** | 2 | 1 | 0 | 3 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | **8** | 0,16 |
| **2** | 0 | 3 | 0 | 5 | 2 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | **16** | 0,33 |
| **3** | 0 | 1 | 1 | 1 | 2 | 1 | 0 | 2 | 0 | 0 | 2 | 0 | 1 | **11** | 0,22 |
| **4** | 1 | 2 | 1 | 0 | 2 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | **11** | 0,22 |
| **5** | 0 | 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | **3** | 0,06 |
| **6** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | **0** | |
| **...** | | | | | | | | | | | | | | | |
| **Encontradas** | 3 | 9 | 2 | 9 | 6 | 5 | 3 | 4 | 1 | 0 | 2 | 2 | 3 | **49** | |
| **De un total** | 12 | 34 | 2 | 18 | 12 | 10 | 9 | 6 | 2 | 12 | 2 | 7 | 7 | | |
| **%** | 0,25 | 0,26 | 1 | 0,5 | 0,5 | 0,5 | 0,33 | 0,67 | 0,5 | 0 | 1 | 0,29 | 0,43 | | |

Cuadro D.1: Número de patrones de jerarquía por nivel de profundidad

| (0): `Entity` |
|---|
| (-1): 4 |
| (-2): 32 |
| (-3): 392 |
| (-4): 2697 |
| (-5): 7747 |
| (-6): 14051 |
| (-7): 26434 |
| (-8): 22529 |
| (-9): 19326 |
| (-10): 16315 |
| (-11): 10244 |
| (-12): 6009 |
| (-13): 3531 |
| (-14): 2089 |
| (-15): 988 |
| (-16): 717 |
| (-17): 356 |
| (-18): 61 |

Cuadro D.2: Frecuencias de términos por nivel en WordNet, a partir de la primera acepción del concepto raíz: `entity`

Al elegir 4 niveles de relaciones jerárquicas descartamos sólo un 6 % de los patrones. Este estudio confirma la segunda consideración respecto a la baja distancia conceptual entre clases. Los patrones, como se verá en posteriores capítulos, sí influyen en la selección de acepciones pero no determinan su idoneidad. Los patrones son relevantes porque causan el aumento del espacio de búsqueda.

# E

# Análisis del uso de clases SPC

En estudio se pretende demostrar y validar la gestión y el uso de las clases SPC (clases estructuralmente predominantes). Como ya hemos comentado, las clases SPC surgen con la idea de simplificar el proceso de descubrimiento de acepciones sin que se merme la calidad. Por tanto, en este análisis compararemos los aciertos usando clases SPC y los aciertos cuando no se usan. Además, aprovechando el estudio, hemos incluido dos cuestiones secundarias. La primera es descubrir cuál es el número óptimo de clases SPC que maximice los aciertos y la segunda, es validar la utilidad de considerar con mayor peso aquellas correspondencias donde participe una clase SPC.

El conjunto de datos corresponde al grupo de ontologías de la categoría de conferencia de AOEI de la campaña 2010. Del total de las 16 ontologías, cuatro son descartadas ya que no se puede aplicar el motor de inferencia cuando tienen anotaciones de OWL-M. Estas cuatro ontologías son: *cocus, confious, iasted* y *linklings*. Las anotaciones de OWL-M sirven para dejar constancia en cada ontología de la acepción de cada palabra mediante los identificadores de WordNet y de *Roget's thesuarus*

El etiquetado de todas las palabras se ha realizado manualmente. Se ha etiquetado un total de 1422 palabras mediante los dos identificadores previamente citados. A la hora de decidir qué palabra o palabras eran las que tenían por si solas significado se ha optado por la prevalencia del significado por encima de su sintaxis, aunque en la mayoría de los casos el formato CamelCase está aplicado correctamente. Es decir, se ha optado por la separación con la que la palabras de la clase tienen un sentido coherente al dominio. Sin embargo, en una serie de casos ha sido imposible definir palabras como: LCD, IASTED, entre otras, por no aparecer en las entradas de ambos tesauros. Además, la versión gratuita de Roget's thesaurus no contiene palabras de nueva cuña de este último siglo o incluso no existe la acepción de la palabra en ese dominio. Algunas palabras de ejemplo son: LCD, CD, monitor, etc. En aquellos casos donde los conceptos no estaban bien escritos se ha procedido a etiquetarlos en base a su forma correcta. Esta serie de palabras mal formadas no son interpretadas por el algoritmo OMoCC ya que no realiza ningún tipo

de corrección o suposición. Se muestra un ejemplo de este etiquetado en el siguiente código E.1.

Código E.1: **Una muestra simplificada del etiquetado de una clase de la ontología cmt**

```
<rdf:Description rdf:about="http://cmt#ProgramCommitteeChair">
  <owlm:iHead rdf:datatype=&rdf:Literal>86 List program
  <owlm:iHead rdf:datatype=&rdf:Literal>696 Council committee
  <owlm:iHead rdf:datatype=&rdf:Literal>694 Director chair
  <owlm:iWordID rdf:datatype=&rdf:Literal>10468962 chair
  <owlm:iWordID rdf:datatype=&rdf:Literal>08324514 committee
  <owlm:iWordID rdf:datatype=&rdf:Literal>06676416 program
```

Basándonos en las definiciones de WordNet existe la posibilidad de aplicar varias acepciones dentro del mismo dominio. Este hecho ha dificultado la selección de una acepción y por consiguiente, la comparación entre acepciones. Varias acepciones pueden ser válidas pero hemos decidido elegir una única acepción. Esto afecta al ratio de aciertos. Una posible solución es la definición de una bondad entre acepciones quedando reflejada en las anotaciones. Ilustramos este caso con los siguientes ejemplos. La palabra `committe` tiene dos acepciones factibles: "a special group delegated to consider some matter" y "a self-constituted organization to promote something". La palabra `paper` tiene 7 acepciones siendo dos de ellas factibles: " an essay (especially one written as an assignment)" y "a scholarly article describing the results of observations or stating hypotheses". La palabra `conference`, de sus tres acepciones dos son más adecuadas: " a discussion among participants who have an agreed (serious) topic" y "a prearranged meeting for consultation or exchange of information or discussion (especially one with a formal agenda)".

## E.1 Evaluación

Con el objetivo de facilitar la compresión de las próximas gráficas hemos elegido una con anotaciones extra para explicar su estructura. De esta manera, la fig. E.1 contiene en la parte superior: el nombre de la ontología, el número total de clases, el número total de clases que contienen nombres compuestos y el número total de palabras derivadas de esas clases, todo ello está en color rojo. La gráfica contiene dos ejes, uno con el número de clases y el otro con el porcentaje de elementos que formarán las clases SPC. En la leyenda, hay cuatro tipos de datos representados. El número de aciertos ($HITS_1$, en color verde) considerando un incremento del peso de las correspondencias cuando participa una clase SCP. El número de aciertos ($HITS_2$, en color ocre) donde todas las clases tienen su peso correspondiente. El número de clases que forman el SPC (*SPC*, en azul). El número de aciertos sin utilizar clases SPC (*NoSPC*, en verde oscuro punteado). Este último valor permanece constante pues no depende del porcentaje. También se visualiza el número de patrones, un número ubicado sobre el número de clases pertenecientes al SPC. El número indica los patrones gracias a las clases SPC. Cuando hay una fracción

‘$\frac{0}{1}$’ el nominador representa los patrones por clases SPC y el denominador los patrones por clases fuera del SPC. El número de patrones (ver 5.3.1) puede variar en función de la aparición de clases que hayan sido descubiertas y de la asignación de una acepción. En la gráfica, con un porcentaje de un 30 % el número de patrones es 1, en cambio en el 40 % existe un patrón más y en el 70 % el número vuelve a decrecer. Los patrones vinculan dos acepciones de diferentes palabras por lo que su aparición descarta ambigüedad pero su no aparición no implica que esas clases tengan asignada una acepción errónea.

Generalmente, el crecimiento de clases SPC no es lineal al porcentaje de clases en total. Este hecho se debe al conjunto de características que comparten grupos de clases entre las diferentes métricas seleccionadas (ver 5.4.1). Un conjunto de clases pueden tener las mismas características bajo un mismo criterio lo que imposibilita que alguna destaque sobre un porcentaje del total a ese criterio. Por ejemplo, cuatro clases pueden tener el mismo número de individuales. Esas 4 clases, en el total superan el umbral de selección de ese criterio por lo que no podrán ser SPC. Cuando el umbral vaya aumentado llegará un momento que esas 4 clases pasen a formar parte del grupo SPC.

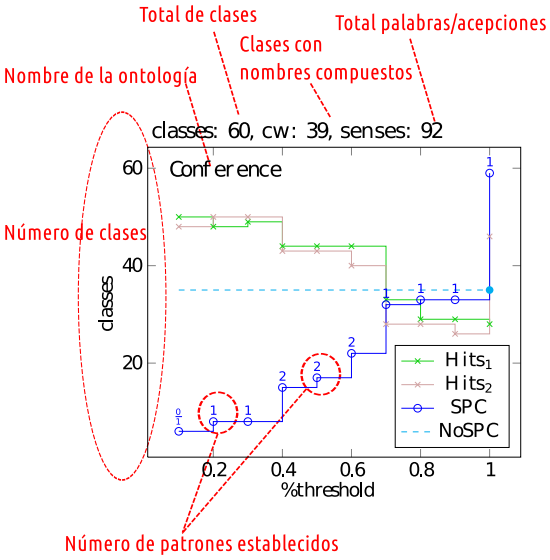


Figura E.1: Gráfica para la explicación de las anotaciones empleadas

Para validar el uso de clases SPC esperamos encontrar en los análisis el siguiente comportamiento: un número reducido de clases SPC obtienen un pico en el número de aciertos y a medida que el número de clases SPC crezca, decrezca el número de aciertos. Este hecho implica que hay un número de clases crítico que define correctamente la acepción del resto de elementos (y de ellos mismos). La ontología `conference`, expuesta en la figura de ejemplo,

tiene el comportamiento esperado. El número de aciertos decrece a medida que el número de conceptos SPC va creciendo. El número de aciertos cuando no se utilizan clases SPC permanece por debajo de su uso (a partir del 70 %). Además, no sería interesante valorar las correspondencias de manera diferente ya que no hay diferencias en los aciertos entre las dos valoraciones: $Hits_1$ y $Hits_2$.

Los resultados se muestran en grupos de 4 para simplificar su exposición. En la figura E.2 se muestran las siguientes ontologías:

- Ontología **conference**. Presenta su máximo de aciertos en el 10 % de SPCs con un 54.34 % respecto a un 38.04 % con no SPCs. El punto significativo de inflexión esta al 70 % con la incorporación de 32 clases del total de 60, aproximadamente al 50 % de las clases el rendimiento de SPC deja de ser útil. No hay diferencias significativas entre las dos valoraciones: $Hits_1$ y $Hits_2$.
- Ontología **OpenConf**. El crecimiento de clases SPC es constante pero da un salto al 70 % de 35 clases. Su máximo de aciertos al 30 % es de 32.18 % respecto a un 21.83 % con no SPCs. La incorporación de más elementos no altera la calidad de los resultados que permanecen constantes.
- Ontología **PCS**. No hay una diferencia entre los aciertos con SPC y no SPC - 3 aciertos de diferencia-, 43.33 % y 33.33 % respectivamente. Existen tres grupos de elementos de SPC en dos porcentajes al 40 % y al 70 %. El número de patrones no afecta a la calidad de los resultados.
- Ontología **Paperdyne**. El número de clases SPC apenas supera 10 de un total de 48 a lo largo de todos los porcentajes exceptuando el 100 %. Al 50 % el número de patrones aumenta y se consigue el número de mayor de aciertos. Al 60 %, existe un 42.85 % de aciertos respecto a 25.97 % cuando no existen clases SPC. Es interesante ver la diferencia entre las dos valoraciones: $Hits_1$ y $Hits_2$, en este caso es mejor no valorar con mayor peso las correspondencias donde participan clases SPC.

En la figura E.3 se muestran las siguientes ontologías:

- Ontología **CRS_DR**. Los elementos que pertenecen a esta ontología no varían con los diferentes porcentajes. Esto se debe a una serie de causas relacionadas con los métricas y por ende causadas por el diseño de la ontología. Esta ontología no tiene ningún individual, ningún axioma, y solo posee un nivel jerárquico entre tres conceptos que comparten el mismo número de hijos entre ellos. A nivel de rango, 9 clases de 14 tienen asignada un rango por lo que no son significativas entre ellas.
- Ontología **edas**. Es la ontología con mayor conceptos analizados 105, aún así el porcentaje de elementos no alcanza el 30 % en todos los porcentajes -exceptuando el 100 %-. El mayor porcentaje de aciertos se alcanza al 20 % con un 41.62 %, pero al 30 % vuelve a decrecer de manera puntual. Cuando se alcanza el 50 % el número de patrones alcanza un número más, 3, consiguiendo un número de aciertos cercanos al número de aciertos de clases no SPC.

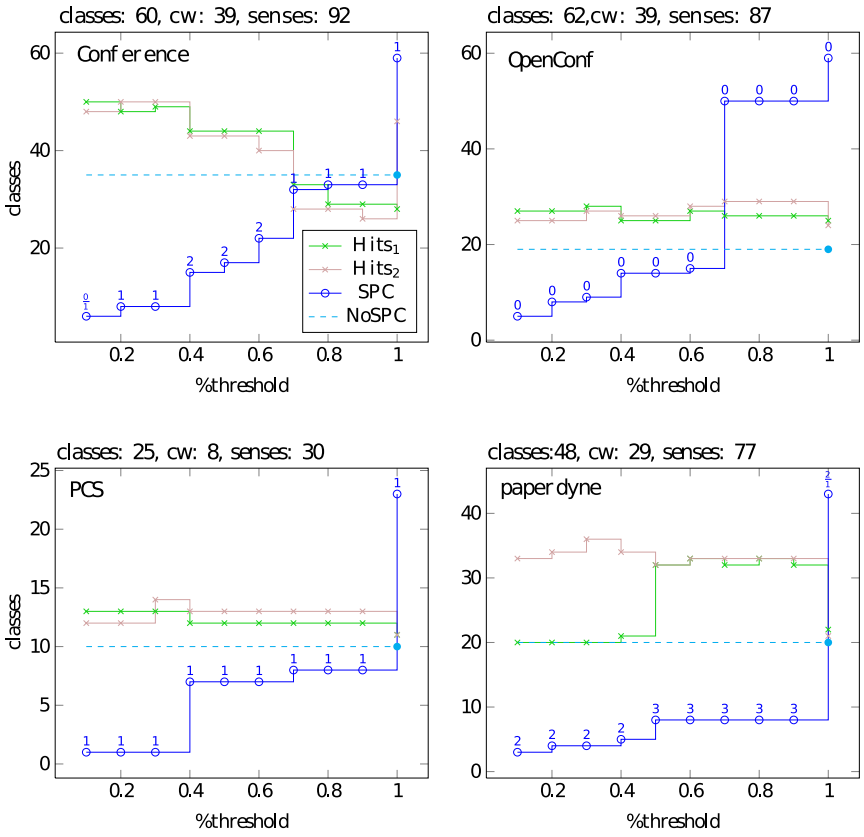


Figura E.2: De izquierda a derecha y de arriba a abajo: *conference, OpenConf, PCS* y *paperdyne*, con sus respectivos porcentajes de aciertos entre clases SPC y no SPC

- Ontología **MICRO**. En *MICRO* el número de aciertos entre ambas propuestas permanece constante e invariable a lo largo de todos los porcentajes exceptuando el 10 % con una diferencia de 1, con un 62 % y un 60 % de aciertos respectivamente. El crecimiento de clases SPC es apenas apreciable con la única salvedad de una perdida de aciertos en los patrones al 50 % que no altera el número de aciertos.
- Ontología **ekaw**. Presenta dos porcentajes de aciertos al 30 % y al 90 % de un 48.73 % y 49.57 % respectivamente, respecto al no utilizar clases SPC que obtienen un 38.65 %. Aparece un hecho ocasional en el 20-30 % ya que aparece un patrón pero no vuelve aparecer al incrementar el porcentaje.

En la figura E.4 se muestran las siguientes ontologías:

- Ontología **CMT**. El porcentaje de aciertos de no utilizar SPC es superior al de utilizarlos, un 54.28 % y 45.71 % respectivamente. El número de pa-

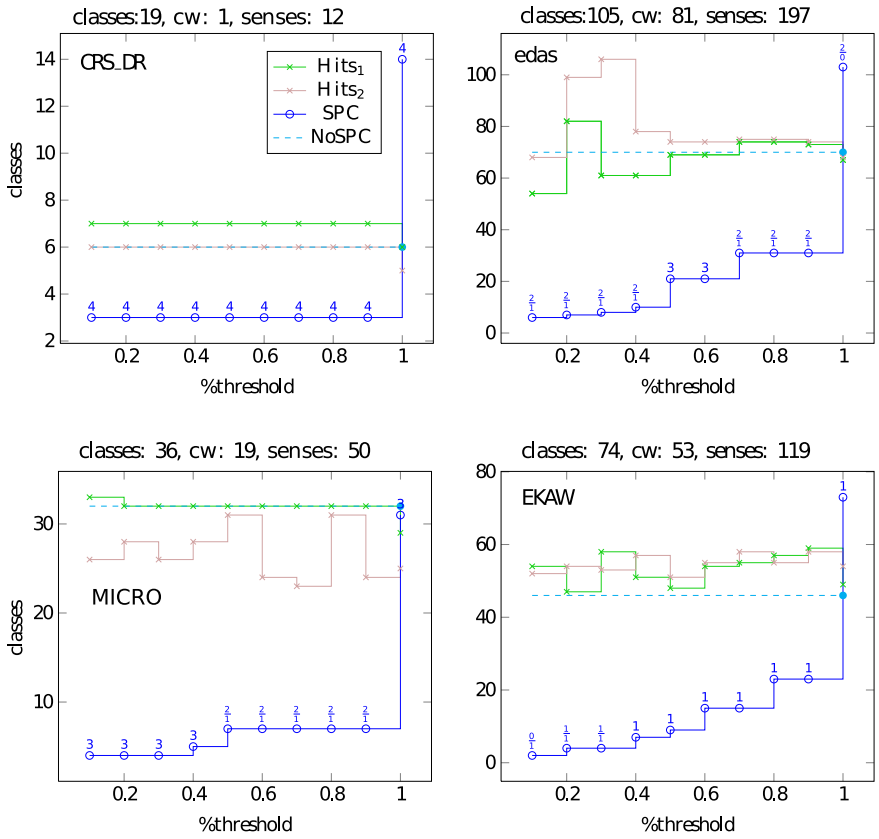


Figura E.3: De izquierda a derecha y de arriba a abajo: CRS_DR, edas, MICRO y ekaw, con sus respectivos porcentajes de aciertos entre clases SPC y no SPC

trones no altera significativamente el número de aciertos. La incorporación de clases al grupo de SPC es creciente pero no vinculante a la calidad de los resultados.

- Ontología **confOf**. En está ontología la diferencia entre usar SPC y no, solo varía en 1 a favor de la última opción, con un éxito de aciertos de 37.77 % y 40 % respectivamente. Además podemos destacar 2 tipos de grupos prácticamente constantes en clases SPC de 10-40 % y 50-90 %. Esta ontología presenta el mayor número de patrones, 8.
- Ontología **SIGKDD**. Con dos máximos de aciertos al uso de SPC en un 10 % y al 50 %, con un 46.42 % de aciertos. El porcentaje de aciertos sin usar clases SPC está en 39.28 %. El número de patrones no altera al número de aciertos pero si que afecta el incremento de clases SPC en el porcentaje de aciertos.

- Ontología **MyReview**. En *MyReview* el porcentaje de aciertos permanece a la par prácticamente desde el 50 %, con un 44.64 % entre ambas ideas. El crecimiento de clases SPC apenas es notable, hay un pequeño incremento en el 50 % pero alcanza 8 de 39 clases posibles.

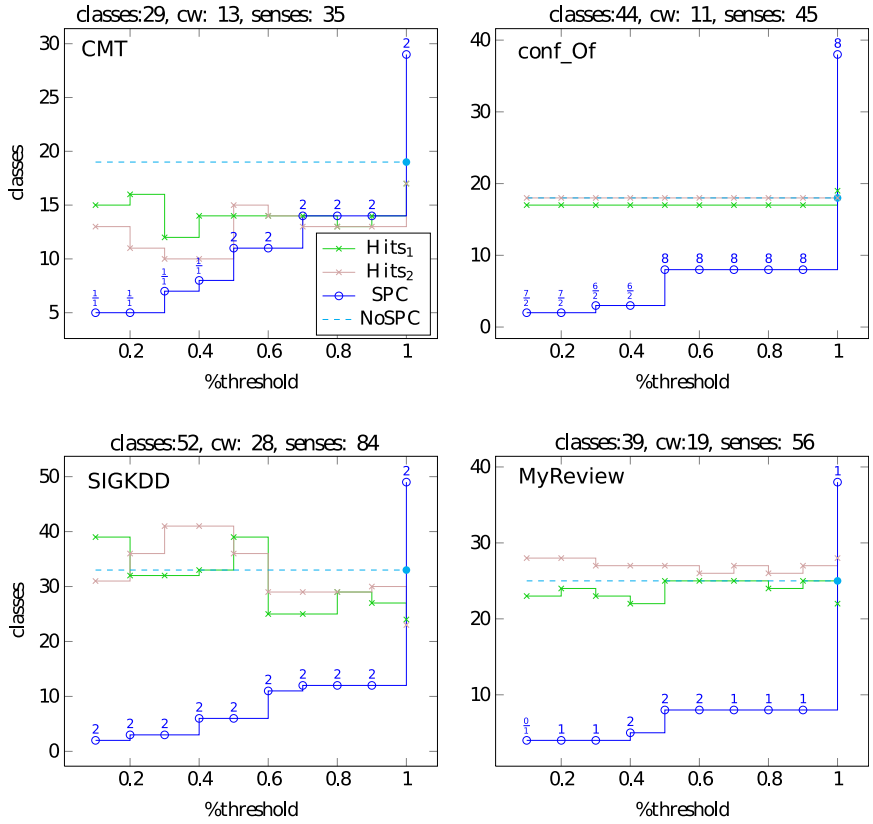


Figura E.4: De izquierda a derecha y de arriba a abajo: *CMT, confOf, SIGKDD* y *MyReview*, con sus respectivos porcentajes de aciertos entre clases SPC y no SPC

### Utilización de recursos computacionales

Hemos realizado un análisis del rendimiento del algoritmo en función del uso y no uso de clases SPC. Las dos métricas utilizadas han sido el tiempo de ejecución y la memoria consumida. La monitorización se realiza para cada ontología en particular. En todas estas pruebas, la aplicación, el sistema y las diferentes valoraciones han sido las mismas.

A nivel del tiempo de ejecución, hemos sintetizado los valores en la fig. E.5. Como se puede apreciar en la leyenda (parte izquierda de la misma) hay una serie de configuraciones del algoritmo.

- SPC: el algoritmo encuentra las clases SPC, descubre sus acepciones y finalmente, descubre las acepciones del resto.
- SPC-*no discover*: el mismo proceso anterior pero donde se resta el tiempo de descubrimiento de clases SPC.
- SPC-*cache*: el algoritmo utiliza clases SPC ya descubiertas, descubre sus acepciones y finalmente, descubre las acepciones del resto.
- NO SPC: el algoritmo descubre las acepciones de todas las clases.

Además de está información, las ontologías están ordenadas por el número de clases. Se incluye el número de propiedades jerárquicas, el número de propiedades entre clases (*object* propiedades), y el número de clases SPC seleccionadas. Aquellas ontologías con un repunte de *object* propiedades presentan tiempos mayores. El tiempo de respuesta es mayor en propuestas con SPC pues hay más llamadas a métodos aunque en ellos el número de recursos sea inferior o igual a la propuestas con NO SPC classes.

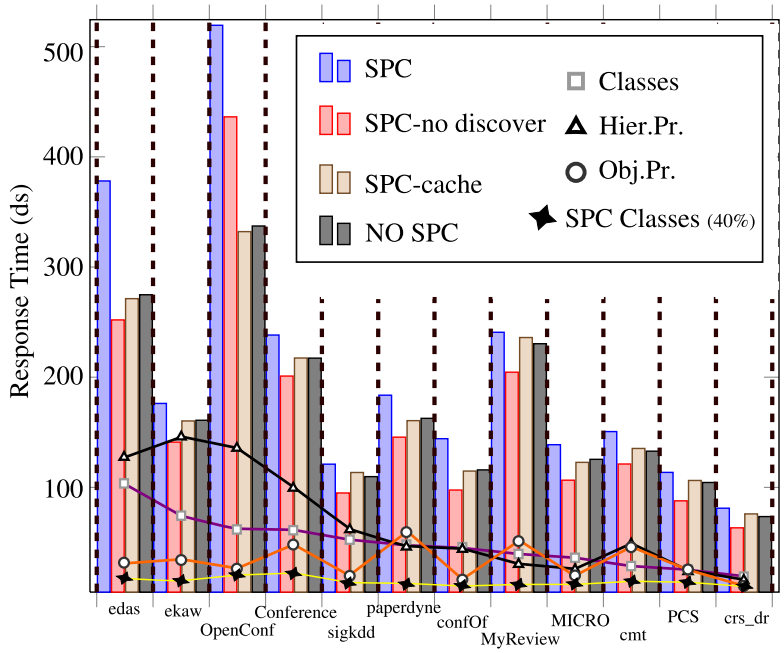


Figura E.5: El tiempo de respuesta (ds) respecto a cada una de las opciones marcadas en la leyenda con sus respectivos datos estructurales

Respecto a la memoria consumida hemos contabilizado los bytes totales de la aplicación para tres casos: uso de clases SPC con el proceso de selección de las mismas, clases SPC con cache, y no uso de clases SPC. Como se puede apreciar en la tabla E.1, la primera configuración presenta el mayor número de bytes utilizados. Hay una ligera mejora en el uso de recursos entre el uso de la cache y el no uso de clases SPC. La aceleración entre ambas propuestas, en la última columna, muestra la pequeña mejora en el uso de clases SPC cuando ya han sido etiquetadas previamente (SPC-cache).

| | SPC | *SPC-cache* | NO SPC | SpeedUp |
|---|---|---|---|---|
| edas | 130.500 | 122.031 | 123.123 | 1.0089492098 |
| ekaw | 56.063 | 53.204 | 52.888 | 0.9940588698 |
| openconf | 275.496 | 174.454 | 173.883 | 0.9967297334 |
| conference | 84.325 | 78.985 | 80.567 | 1.0200315513 |
| sigkdd | 25.472 | 23.409 | 23.702 | 1.0125045221 |
| paperdyne | 41.126 | 36.838 | 37.432 | 1.0161489214 |
| confof | 29.798 | 22.911 | 23.100 | 1.0082143619 |
| myreview | 94.613 | 92.280 | 92.131 | 0.9983865622 |
| micro | 33.620 | 30.222 | 31.246 | 1.0339106494 |
| cmt | 31.373 | 29.660 | 29.209 | 0.9847929235 |
| pcs | 22.218 | 20.481 | 20.587 | 1.0051677748 |
| crs_dr | 12.209 | 12.021 | 11.562 | 0.9617488732 |
| | | | Total | 1.0033869961 |

Cuadro E.1: Memoria consumida (MB) en la averiguación de clases SPC, *SPC-cache*, NO SPC y la aceleración conseguida respecto a las dos últimas

## E.2 Comentarios generales

La búsqueda de patrones permite relacionar acepciones entre conceptos mediante la comparación de ambas jerarquías. Estos patrones fijan las acepciones de esas dos clases. En el estudio vemos que el número de patrones no afecta a los aciertos. Como ya hemos comentado, esto se debe a varias razones: hay más de una única acepción posible pero nosotros sólo gestionamos una; el diseño de la propia ontología con el uso adecuado de constructores y por último, la granularidad y diseño de WordNet. WordNet [108] no fue diseñado para tareas de desambiguación y al ser la mejor opción disponible se convirtió en el recurso más utilizado para este propósito. Las aplicaciones de desambiguación sufren extremadamente las distinciones de matices que caracterizan a una palabra en WordNet [1], es decir, el nivel de granularidad en la definición, y en otros casos la escasez de palabras que complementan una acepción en su contexto [107]. En [1] múltiples soluciones son dadas desde “WordNet” específicos

a dominios, el incremento de información preferencial y de frecuencia, simplificación de jerarquías, acepciones mutuamente exclusivas, eliminando enlaces incorrectos, entre otras propuestas.

Hay una serie de ontologías (paperdyne, CRS_DR, MICRO y MyReview) donde el crecimiento de clases SPC es apenas significante. Este crecimiento es nulo en CRS_DR por el contrario EKAW presenta un crecimiento ideal. La diferencia entre ambas está en el repertorio de estructuras jerárquicas, axiomas, rangos y en los individuales. Ambos diseños son diferentes y ambos cumplen con los requisitos de su aplicación pero es evidente que una ontología emplea menos constructores semánticos con lo que presenta un repertorio taxonómico y axiomático más pobre. Sin embargo, el acierto de acepciones es similar a ambas: CRS_DR presenta un 50 % y EKAW, 49.57 %.

El número de pruebas para este estudio está limitado al grupo de ontologías que tienen sus acepciones anotadas, siendo en este caso 12 de las 16 posibles. Estas cuatro ontologías son descartadas a causa del motor de inferencia (Pellet-2.0.0-rc4) que no puede manipular de manera eficiente las anotaciones de acepciones. Estadísticamente, el juego de pruebas es escaso pero hay que tener en cuenta la dificultad de encontrar ontologías de dominios similares y adecuadamente desarrolladas. La iniciativa AOEI engloba y emplea ontologías con un alto grado de descripción. Esta selección de ontologías son las más propicias para este tipo de funciones pero sería necesario incluir más casos.

En la fig. E.6 vemos una comparativa ordenada de aciertos de cada ontología. Corresponden a los aciertos $Hits_2$. El máximo de aciertos (*max. hits*) corresponde al umbral del clases SPC que genera el mayor número de aciertos, en cada caso es diferente. El número de aciertos (*hits*) corresponde al 30 % de clases SPC. Se puede apreciar que no hay ninguna relación directa entre el porcentaje de aciertos y el tamaño de la ontología. La media de aciertos se mantiene alrededor del 50 %.

La fig. E.7 contiene la síntesis de todos los anteriores casos. Los valores están normalizados en función del número de acepciones. Se aprecia el aumento de las clases SPC a medida que aumenta el porcentaje de selección. El número de aciertos con SPC es ligeramente superior al número de aciertos sin SPC. También, el número de aciertos $Hits_2$, al no considerar correspondencias con pesos diferentes cuando hay clases SPC, ofrece mejores resultados que cuando se hace tal distinción $Hits_1$. Es ligeramente decreciente el número de elementos de SPC. El pico de aciertos con $Hits_1$ se produce al 50 % con 43.72 %. El pico de aciertos con $Hits_2$ se produce al 30 % con 45.39 %. Al 50 % de elementos SPC se establecen el mayor número de patrones. Al 50 % de clases SPC el número de elementos que pertenecen a este grupo es de 20.90 %, y al 30 %, el número de clases SPC es de 10.59 %.

La fig. E.8 contiene una comparativa de aciertos entre tres propuestas: el no uso de clases SPC, y su uso con diferentes porcentajes de clases SPC elegidas (al 30 % y al 40 %). En los tres casos se ha utilizado el mismo conjunto de valoraciones de correspondencias: el de frecuencias. Además, ha obtenido

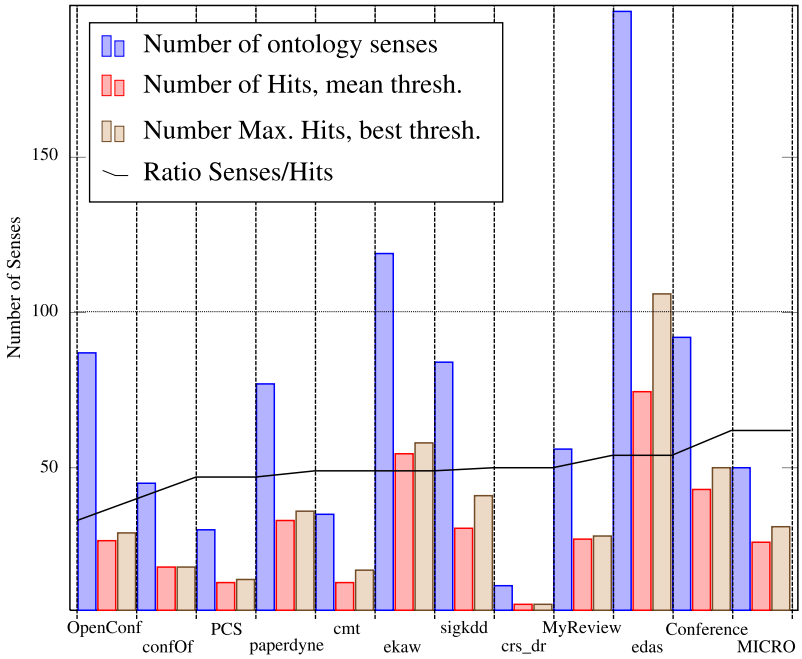


Figura E.6: Comparativa ordenada de aciertos ponderados según cada ontología

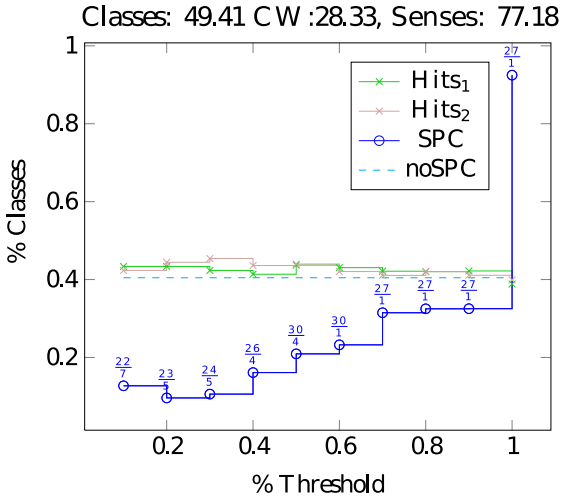


Figura E.7: Valores normalizados de todas las ontologías, con sus respectivos porcentajes de aciertos entre clases SPC y no SPC

los mejores resultados. La diferencia entre etiquetado al 40 % y al no uso de clases SPC es sensiblemente mejor pero sin una gran diferencia notable: en seis casos es significativo, en un caso es idéntico y en 5 es ligeramente menor.

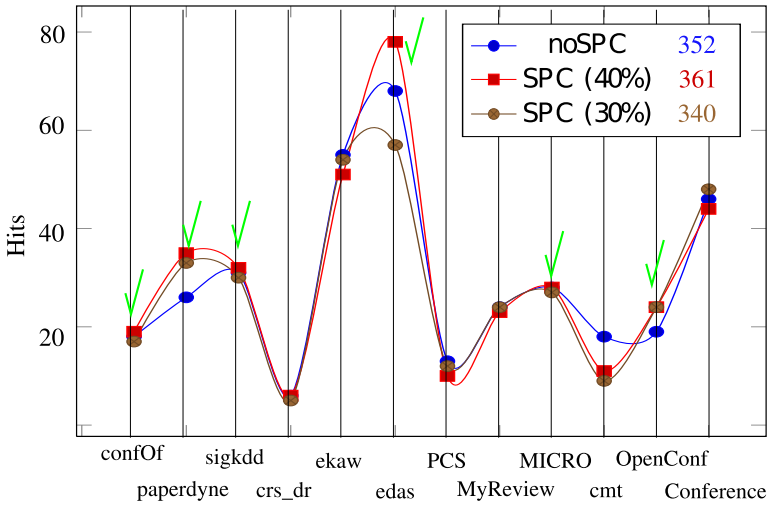


Figura E.8: Comparativa de aciertos entre usar clases SPC y no usarlas bajo medidas de frecuencia

Las clases SPC pueden ser etiquetadas durante la fase de creación de la ontología y ser utilizadas en cualquier aplicación sin necesidad de realizar el proceso de búsqueda de las mismas. De esta manera, el consumo de recursos se ve reducido ligeramente.

La conclusión es que existe un número significativo de clases que por los motivos comentados delimitan el contexto del resto de elementos. Este grupo de elementos no sobrepasa el 40 % del total de elementos de la ontología. Esta idea puede ser utilizada en sistemas donde el rendimiento sea importante , en cuestiones de caché o interpretación rápida de la representación global de la ontología.

# F

# Análisis de la presencia de nombres compuestos

En este anexo se ha realizado un estudio para contabilizar el número de nombres compuestos presentes en la clases. El estudio se ha realizado con el repertorio de ontologías de OAEI `conference` de 2010. Los resultados se muestran en la tabla F.1. En la segunda y tercera columna, se representa el total de nombres compuestos con o sin significado, respectivamente. El porcentaje de nombres compuestos respecto al total de clases está representado en la última columna.

| | Clases | Compuestas: *con* | Compuestas: *sin* | % |
|---|---|---|---|---|
| cmt | 32 | 1 | 17 | 0,56 |
| cocus | 70 | 0 | 27 | 0,39 |
| conference | 63 | 2 | 51 | 0,84 |
| confOf | 44 | 2 | 13 | 0,34 |
| crs_dr | 25 | 0 | 1 | 0,04 |
| edas | 111 | 0 | 94 | 0,85 |
| ekaw | 77 | 2 | 57 | 0,77 |
| micro | 39 | 1 | 21 | 0,56 |
| myreview | 39 | 1 | 22 | 0,59 |
| openconf | 62 | 2 | 47 | 0,79 |
| paperdyne | 48 | 3 | 33 | 0,75 |
| pcs | 28 | 2 | 11 | 0,46 |
| sigkdd | 55 | 1 | 29 | 0,55 |
| | | | **Total** | **0,56** |

Cuadro F.1: Número de clases compuestas con y sin significado

Tal como se puede apreciar en este análisis el número de clases con nombres compuestos es significativo, alcanzando una media de 56 %. Por este motivo es importante considerar y gestionar el aporte de estos términos al contexto de la representación.

# G

# Ontologías para un caso de estudio

En este anexo se muestran las ontologías usadas en el capítulo de evaluación. Pueden ser accesibles mediante su correspondiente URI.

Código de la ontología: *OntoAuthor.owl*.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns="http://acsic.uib.es/OntoAuthor2.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://acsic.uib.es/OntoAuthor2.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="Book">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Creation"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="CollectionBook">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Collection"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="Author"/>
  <owl:Class rdf:ID="CollectionCD">
    <rdfs:subClassOf rdf:resource="#Collection"/>
  </owl:Class>
  <owl:Class rdf:ID="CD">
    <rdfs:subClassOf rdf:resource="#Creation"/>
  </owl:Class>
  <owl:ObjectProperty rdf:ID="do">
    <rdfs:domain rdf:resource="#Author"/>
    <rdfs:range rdf:resource="#Book"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="has">
    <rdfs:range rdf:resource="#Creation"/>
    <rdfs:domain rdf:resource="#Collection"/>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:about="http://acsic.uib.es/OntoAuthor.owl#
      hasPages">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#int"/>
    <rdfs:domain rdf:resource="#Book"/>
  </owl:DatatypeProperty>
</rdf:RDF>
```

Código de la ontología: *OntoWriter.owl*.

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://acsic.uib.es/OntoWriter2.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://acsic.uib.es/OntoWriter2.owl">
  <owl:Ontology rdf:about=""/>
  <rdfs:Class rdf:ID="Book">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Work"/>
    </rdfs:subClassOf>
  </rdfs:Class>
  <owl:Class rdf:ID="Collection"/>
  <owl:Class rdf:ID="Writer"/>
  <owl:ObjectProperty rdf:ID="isAuthor">
    <rdfs:domain rdf:resource="#Book"/>
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="write"/>
    </owl:inverseOf>
    <rdfs:range rdf:resource="#Writer"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#write">
    <rdfs:range rdf:resource="#Book"/>
    <rdfs:domain rdf:resource="#Writer"/>
    <owl:inverseOf rdf:resource="#isAuthor"/>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="belong">
    <rdfs:domain rdf:resource="#Work"/>
    <rdfs:range rdf:resource="#Collection"/>
  </owl:ObjectProperty>
</rdf:RDF>
```

Código de la unión de ambas ontologías: *OntoAuthor* y *OntoWriter*, mediante OWL-M.

```
<rdf:RDF
    xmlns:Model1="http://acsic.uib.es/OntoAuthor2.owl#"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns:Model2="http://acsic.uib.es/OntoWriter2.owl#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns="http://acsic.uib.es/OntoWriter2OntoAuthor2.owl#"
    xmlns:owlm="http://swap.uib.es/2009/08/owlm#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
  <rdf:Description rdf:about="http://acsic.uib.es/OntoAuthor2.owl#
      CollectionBook">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#Book
        "/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/OntoAuthor2.owl#Book">
    <owlm:similarOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#Book"
        />
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/OntoAuthor2.owl#
        product"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/OntoWriter2OntoAuthor2.
      owl#">
    <owl:imports rdf:resource="file:///home/isaac/Escritorio/testDataSet/
        twoOntos2/OntoWriter2.owl"/>
    <owl:imports rdf:resource="file:///home/isaac/Escritorio/testDataSet/
        twoOntos2/OntoAuthor2.owl"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>
  </rdf:Description>
```

```
  <rdf:Description rdf:about="http://acsic.uib.es/OntoWriter2.owl#
      publication">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#Work
        "/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/OntoAuthor2.owl#Author">
    <owlm:hypernymOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#
        Collection"/>
    <owlm:hypernymOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#
        Writer"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/OntoAuthor2.owl#product">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/OntoAuthor2.owl#
        Creation"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/OntoAuthor2.owl#
      Collection">
    <owlm:similarOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#
        Collection"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://swap.uib.es/2009/08/owlm#product">
    <owlm:actionRule rdf:resource="http://acsic.uib.es/OntoWriter2.owl#Work
        "/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/OntoWriter2.owl#Book">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/OntoWriter2.owl#
        publication"/>
  </rdf:Description>
</rdf:RDF>
```

Código de la unión de ambas ontologías: *OntoAuthor* y *OntoWriter*, mediante Alignment API.

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>
  <rdf:RDF xmlns='http://knowledgeweb.semanticweb.org/heterogeneity/
      alignment#'
    xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'
    xmlns:xsd='http://www.w3.org/2001/XMLSchema#'
    xmlns:align='http://knowledgeweb.semanticweb.org/heterogeneity/
        alignment#'>
  <Alignment>
  <xml>yes</xml>
  <level>0</level>
  <type>**</type>
 <onto1><Ontology rdf:about=".../OntoAuthor2.owl">
  <location>file:/.../OntoAuthor2.owl</location></Ontology></onto1>
 <onto2><Ontology rdf:about=".../OntoWriter2.owl">
 <location>file:/.../OntoWriter2.owl</location></Ontology></onto2>

 <map>
 <Cell>
  <entity1 rdf:resource='http://acsic.uib.es/OntoAuthor2.owl#Collection'/>
  <entity2 rdf:resource='http://acsic.uib.es/OntoWriter2.owl#Collection'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>

  <map>
 <Cell>
  <entity1 rdf:resource='http://acsic.uib.es/OntoAuthor2.owl#Book'/>
  <entity2 rdf:resource='http://acsic.uib.es/OntoWriter2.owl#Book'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
```

```
</map>
 </Alignment>
 </rdf:RDF>
```

Código de la unión de *EDAS* y *EKAW*, mediante OWL-M.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://acsic.uib.es/ncB.owl#"
    xmlns:Model1="http://edas#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:owlm="http://swap.uib.es/2009/08/owlm#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:Model2="http://ekaw#" >
  <rdf:Description rdf:about="http://ekaw#Research_Topic">
    <rdfs:subClassOf rdf:resource="http://edas#Topic"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#MealEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#communicator">
    <rdfs:subClassOf rdf:resource="http://edas#Person"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Person">
    <owlm:actionRule rdf:resource="http://ekaw#Organisation"/>
    <owlm:actionRule rdf:resource="http://swap.uib.es/2009/08/owlm#enrollee
        "/>
    <owl:equivalentClass rdf:resource="http://ekaw#Person"/>
    <owlm:actionRule rdf:resource="http://ekaw#Social_Event"/>
    <owlm:actionRule rdf:resource="http://ekaw#Presenter"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Evaluated_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Camera_Ready_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Poster_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Individual_Presentation">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#presentation
        "/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Conference_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#ConferenceSession">
    <owlm:synonymOf rdf:resource="http://ekaw#Conference_Session"/>
    <rdfs:subClassOf rdf:resource="http://ekaw#Session"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#ClosingTalk">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Paper">
    <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
    <owl:equivalentClass rdf:resource="http://ekaw#Paper"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
```

```
</rdf:Description>
<rdf:Description rdf:about="http://edas#NonAcademicEvent">
  <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#writer">
  <rdfs:subClassOf rdf:resource="http://edas#Person"/>
  <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Tutorial_Chair">
  <rdfs:subClassOf rdf:resource="http://ekaw#SessionChair"/>
  <rdfs:subClassOf rdf:resource="http://ekaw#ConferenceChair"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#CallForReviews">
  <rdfs:subClassOf rdf:resource="http://ekaw#Review"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Topic">
  <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Industrial_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#WelcomeTalk">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#enrollee">
  <rdfs:subClassOf rdf:resource="http://ekaw#Person"/>
</rdf:Description>
<rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#chair">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#AcademicEvent">
  <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#PaperPresentation">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#presentation
      "/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Neutral_Review">
  <rdfs:subClassOf rdf:resource="http://edas#Review"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#SingleLevelConference">
  <rdfs:subClassOf rdf:resource="http://ekaw#Conference"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Paper_Author">
  <rdfs:subClassOf rdf:resource="http://edas#Author"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#PC_Chair">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Place">
  <owlm:actionRule rdf:resource="http://ekaw#Location"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Document">
  <owl:equivalentClass rdf:resource="http://ekaw#Document"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Demo_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
```

```
<rdf:Description rdf:about="http://edas#Conference">
  <owlm:similarOf rdf:resource="http://ekaw#Conference"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#PublishedPaper">
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#PendingPaper">
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#ConferenceEvent">
  <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#ActivePaper">
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#SessionChair">
  <owlm:synonymOf rdf:resource="http://ekaw#Session_Chair"/>
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#SocialEvent">
  <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#TwoLevelConference">
  <rdfs:subClassO<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:owl="http://www.w3.org/2002/07/owl#"
      xmlns="http://acsic.uib.es/ncB.owl#"
      xmlns:Model1="http://edas#"
      xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
      xmlns:owlm="http://swap.uib.es/2009/08/owlm#"
      xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
      xmlns:Model2="http://ekaw#" >
    <rdf:Description rdf:about="http://ekaw#Research_Topic">
      <rdfs:subClassOf rdf:resource="http://edas#Topic"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://edas#MealEvent">
      <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://edas#communicator">
      <rdfs:subClassOf rdf:resource="http://edas#Person"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://edas#Person">
      <owlm:actionRule rdf:resource="http://ekaw#Organisation"/>
      <owlm:actionRule rdf:resource="http://swap.uib.es/2009/08/owlm#
          enrollee"/>
      <owl:equivalentClass rdf:resource="http://ekaw#Person"/>
      <owlm:actionRule rdf:resource="http://ekaw#Social_Event"/>
      <owlm:actionRule rdf:resource="http://ekaw#Presenter"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://ekaw#Evaluated_Paper">
      <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://ekaw#Camera_Ready_Paper">
      <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://ekaw#Poster_Paper">
      <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://ekaw#Individual_Presentation">
      <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#
          presentation"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://ekaw#Conference_Paper">
      <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://edas#ConferenceSession">
      <owlm:synonymOf rdf:resource="http://ekaw#Conference_Session"/>
```

```
    <rdfs:subClassOf rdf:resource="http://ekaw#Session"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#ClosingTalk">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Paper">
    <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
    <owl:equivalentClass rdf:resource="http://ekaw#Paper"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#NonAcademicEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#writer">
    <rdfs:subClassOf rdf:resource="http://edas#Person"/>
    <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Tutorial_Chair">
    <rdfs:subClassOf rdf:resource="http://ekaw#SessionChair"/>
    <rdfs:subClassOf rdf:resource="http://ekaw#ConferenceChair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#CallForReviews">
    <rdfs:subClassOf rdf:resource="http://ekaw#Review"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Topic">
    <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Industrial_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#WelcomeTalk">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#enrollee">
    <rdfs:subClassOf rdf:resource="http://ekaw#Person"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#chair">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#AcademicEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#PaperPresentation">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#
        presentation"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Neutral_Review">
    <rdfs:subClassOf rdf:resource="http://edas#Review"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#SingleLevelConference">
```

```
    <rdfs:subClassOf rdf:resource="http://ekaw#Conference"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Paper_Author">
    <rdfs:subClassOf rdf:resource="http://edas#Author"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#PC_Chair">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Place">
    <owlm:actionRule rdf:resource="http://ekaw#Location"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Document">
    <owl:equivalentClass rdf:resource="http://ekaw#Document"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Demo_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Conference">
    <owlm:similarOf rdf:resource="http://ekaw#Conference"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#PublishedPaper">
    <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#PendingPaper">
    <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#ConferenceEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#ActivePaper">
    <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#SessionChair">
    <owlm:synonymOf rdf:resource="http://ekaw#Session_Chair"/>
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#SocialEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#TwoLevelConference">
    <rdfs:subClassOf rdf:resource="http://ekaw#Conference"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#">
    <owl:imports rdf:resource="file:///home/isaac/Escritorio/
        testDataSet/ConferenceOriginales/ekaw.owl"/>
    <owl:imports rdf:resource="file:///home/isaac/Escritorio/
        testDataSet/ConferenceOriginales/edas.owl"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#TalkEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Assigned_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#BreakEvent">
    <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Student">
    <rdfs:subClassOf rdf:resource="http://ekaw#enrollee"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Session_Chair">
    <rdfs:subClassOf rdf:resource="http://ekaw#ConferenceChair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Accepted_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
```

```
<rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#talk">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Author">
  <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
  <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#OC_Chair">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Possible_Reviewer">
  <rdfs:subClassOf rdf:resource="http://edas#Reviewer"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#WithdrawnPaper">
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Organization">
  <owlm:synonymOf rdf:resource="http://ekaw#Organisation"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Workshop_Chair">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#ConferenceChair">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Negative_Review">
  <rdfs:subClassOf rdf:resource="http://edas#Review"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Rejected_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#AcceptedPaper">
  <owlm:synonymOf rdf:resource="http://ekaw#Accepted_Paper"/>
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Reviewer">
  <rdfs:subClassOf rdf:resource="http://edas#writer"/>
  <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Contributed_Talk">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Regular_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Invited_Talk">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Workshop_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Demo_Chair">
  <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#RejectedPaper">
  <owlm:synonymOf rdf:resource="http://ekaw#Rejected_Paper"/>
  <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
</rdf:Description>
```

```
    <rdf:Description rdf:about="http://edas#Presenter">
      <owlm:similarOf rdf:resource="http://ekaw#Presenter"/>
      <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://ekaw#Positive_Review">
      <rdfs:subClassOf rdf:resource="http://edas#Review"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#presentation"
        >
      <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://edas#Review">
      <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
      <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
      <owlm:similarOf rdf:resource="http://ekaw#Review"/>
    </rdf:Description>
    <rdf:Description rdf:about="http://edas#Workshop">
      <owl:equivalentClass rdf:resource="http://ekaw#Workshop"/>
    </rdf:Description>
  </rdf:RDF>f rdf:resource="http://ekaw#Conference"/>
</rdf:Description>
<rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#">
  <owl:imports rdf:resource="file:///home/isaac/Escritorio/testDataSet/
      ConferenceOriginales/ekaw.owl"/>
  <owl:imports rdf:resource="file:///home/isaac/Escritorio/testDataSet/
      ConferenceOriginales/edas.owl"/>
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Ontology"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#TalkEvent">
  <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Assigned_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#BreakEvent">
  <rdfs:subClassOf rdf:resource="http://ekaw#Event"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Student">
  <rdfs:subClassOf rdf:resource="http://ekaw#enrollee"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Session_Chair">
  <rdfs:subClassOf rdf:resource="http://ekaw#ConferenceChair"/>
</rdf:Description>
<rdf:Description rdf:about="http://ekaw#Accepted_Paper">
  <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
</rdf:Description>
<rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#talk">
  <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
</rdf:Description>
<rdf:Description rdf:about="http://edas#Author">
  <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
  <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
  <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
```

```
    <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#OC_Chair">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Possible_Reviewer">
    <rdfs:subClassOf rdf:resource="http://edas#Reviewer"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#WithdrawnPaper">
    <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Organization">
    <owlm:synonymOf rdf:resource="http://ekaw#Organisation"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Workshop_Chair">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#ConferenceChair">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Negative_Review">
    <rdfs:subClassOf rdf:resource="http://edas#Review"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Rejected_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#AcceptedPaper">
    <owlm:synonymOf rdf:resource="http://ekaw#Accepted_Paper"/>
    <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Reviewer">
    <rdfs:subClassOf rdf:resource="http://edas#writer"/>
    <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Contributed_Talk">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Regular_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Invited_Talk">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#talk"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Workshop_Paper">
    <rdfs:subClassOf rdf:resource="http://edas#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Demo_Chair">
    <rdfs:subClassOf rdf:resource="http://acsic.uib.es/ncB.owl#chair"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#RejectedPaper">
    <owlm:synonymOf rdf:resource="http://ekaw#Rejected_Paper"/>
    <rdfs:subClassOf rdf:resource="http://ekaw#Paper"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Presenter">
    <owlm:similarOf rdf:resource="http://ekaw#Presenter"/>
    <rdfs:subClassOf rdf:resource="http://edas#communicator"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ekaw#Positive_Review">
    <rdfs:subClassOf rdf:resource="http://edas#Review"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://acsic.uib.es/ncB.owl#presentation">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Class"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Review">
    <owlm:hypernymOf rdf:resource="http://ekaw#Document"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Organisation"/>
```

```
    <owlm:hypernymOf rdf:resource="http://ekaw#Presenter"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Proceedings"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Workshop"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#University"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Conference"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Track"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Flyer"/>
    <owlm:hypernymOf rdf:resource="http://ekaw#Location"/>
    <owlm:similarOf rdf:resource="http://ekaw#Review"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://edas#Workshop">
    <owl:equivalentClass rdf:resource="http://ekaw#Workshop"/>
  </rdf:Description>
</rdf:RDF>
```

Código de la unión de *EDAS* y *EKAW*, mediante Alignment API.

```
<?xml version='1.0' encoding='utf-8' standalone='no'?>  <rdf:RDF xmlns='
    http://knowledgeweb.semanticweb.org/heterogeneity/alignment#' xmlns:
    rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#' xmlns:xsd='http://
    www.w3.org/2001/XMLSchema#' xmlns:align='http://knowledgeweb.
    semanticweb.org/heterogeneity/alignment#'> <Alignment>   <xml>yes</xml
    >    <level>0</level>    <type>**</type>
 <onto1><Ontology rdf:about="http://seals.sti2.at/tdrs-web/testdata/
     persistent/Conference+Testsuite/2010/suite/edas-ekaw/component/source
     /">  <location>http://seals.sti2.at/tdrs-web/testdata/persistent/
     Conference+Testsuite/2010/suite/edas-ekaw/component/source/</location
     ></Ontology></onto1>
 <onto2><Ontology rdf:about="http://seals.sti2.at/tdrs-web/testdata/
     persistent/Conference+Testsuite/2010/suite/edas-ekaw/component/target
     /">  <location>http://seals.sti2.at/tdrs-web/testdata/persistent/
     Conference+Testsuite/2010/suite/edas-ekaw/component/target/</location
     ></Ontology></onto2>
 <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Review'/>
  <entity2 rdf:resource='http://ekaw#Review'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>
  <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Person'/>
  <entity2 rdf:resource='http://ekaw#Person'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>
  <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Paper'/>
  <entity2 rdf:resource='http://ekaw#Paper'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>
  <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Document'/>
  <entity2 rdf:resource='http://ekaw#Document'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
```

```
</Cell>
</map>
  <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Workshop'/>
  <entity2 rdf:resource='http://ekaw#Workshop'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>
  <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Presenter'/>
  <entity2 rdf:resource='http://ekaw#Presenter'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>
  <map>
 <Cell>
  <entity1 rdf:resource='http://edas#Conference'/>
  <entity2 rdf:resource='http://ekaw#Conference'/>
  <relation>=</relation>
  <measure rdf:datatype='http://www.w3.org/2001/XMLSchema#float'>1</measure
      >
</Cell>
</map>
  </Alignment></rdf:RDF>
```

## Referencias


[1] Eneko Agirre and Philip Glenny Edmonds. *Word Sense Disambiguation: Algorithms and Applications*. Springer, 2006.

[2] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010.

[3] Yuan An and Il-Yeol Song. Discovering Semantically Similar Associations (SeSA) for Complex Mappings between Conceptual Models. In *Proceedings of the 27th International Conference on Conceptual Modeling*, ER '08, pages 369–382, Berlin, Heidelberg, 2008. Springer-Verlag.

[4] Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer, 2nd Edition (Cooperative Information Systems)*. The MIT Press, 2 edition, 2008.

[5] David Aumueller, Hong-Hai Do, Sabine Massmann, and Erhard Rahm. Schema and ontology matching with COMA++. In *IN SIGMOD CONFERENCE*, pages 906–908. ACM Press, 2005.

[6] Marko Bane, Boris Vrdoljak, and A. Min Tjoa. Word sense disambiguation as the primary step of ontology integration. In *Proceedings of the 19th international conference on Database and Expert Systems Applications*, DEXA '08, pages 65–72, Berlin, Heidelberg, 2008. Springer-Verlag.

[7] Jie Bao, Jiao Tao, Deborah L. McGuinness, and Paul R. Smart. Context representation for the semantic web. In *Web Science Conference*, 2010.

[8] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. In *International AAAI Conference on Weblogs and Social Media*, 2009.

[9] Chris Biemann and Valerie Nygaard. Crowd-sourcing WordNet. In *Proceedings of the 5th Global WordNet conference*, Mumbai, India, 2010. ACL Data and Code Repository.

[10] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, March 2003.

[11] Jürgen Bock, Alexander Lenk, and Carsten Dänschel. Ontology alignment in the cloud. In *Proceedings of the Fifth International Workshop on Ontology Matching (OM2010), Shanghai, China*. CEUR-WS, November 2010.

[12] Jürgen Bock, Peng Liu, and Jan Hettenhausen. Mappso results for oaei 2009. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Natalya Fridman Noy, and Arnon Rosenthal, editors, *Ontology Matching*, volume 551 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[13] Philip Bohannon, Eiman Elnahrawy, Wenfei Fan, and Michael Flaster. Putting context into schema matching. In *Proceedings of the 32nd international conference on Very large data bases*, VLDB '06, pages 307–318. VLDB Endowment, 2006.

[14] Elena Paslaru Bontas. Context representation and usage for the semantic web: A state of the art. Technical Report B-04-30, Freie Universität Berlin, 2004.

[15] Stéphane Bressan, Akmal B. Chaudhri, Mong-Li Lee, Jeffrey Xu Yu, and Zoé Lacroix, editors. *Efficiency and Effectiveness of XML Tools and Techniques and Data Integration over the Web, VLDB 2002 Workshop EEXTT and CAiSE 2002 Workshop DTWeb. Revised Papers*, volume 2590 of *Lecture Notes in Computer Science*. Springer, 2003.

[16] Alexander Budanitsky and Graeme Hirst. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computacional Linguistics*, 32(1):13–47, March 2006.

[17] Mario Bunge. *Treatise on Basic Philosophy: Volume 1: Semantics I: Sense and Reference*. Springer, 1974.

[18] Mario Bunge. *Treatise on Basic Philosophy: Volume 2: Semantics II: Interpretation and Truth*. Springer, 1974.

[19] Fayçal Hamdi, Haïfa Zargayouna, Brigitte Safar, and Chantal Reynaud. Taxomap in the oaei 2008 alignment contest. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Heiner Stuckenschmidt, editors, *OM*, volume 431 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[20] Lucie Marie Le Cam and Grace Lo Yang. *Asymptotics in statistics: some basic concepts*. Springer series in statistics. Springer-Verlag, 2nd edition, 2000.

[21] S. Castano, A. Ferrara, and S. Montanelli. H-match: an algorithm for dynamically matching ontologies in peer-based systems. In *In Proc. of the 1st Int. Workshop on Semantic Web and Databases (SWDB) at VLDB 2003*, pages 231–250, 2003.

[22] N. Choi, I.Y. Song, and H. Han. A survey on ontology mapping. *ACM Sigmod Record*, 35(3):34–41, 2006.

[23] Watson Wei Khong Chua and Jung-Jae Kim. Eff2match results for oaei 2010. In Shvaiko et al. [143].

[24] William F. Clocksin and Christopher S. Mellish. *Programming in Prolog: Using the ISO Standard*. Springer, 5th edition, September 2003.

[25] William W. Cohen, Pradeep Ravikumar, and Stephen E. Fienberg. A Comparison of String Distance Metrics for Name-Matching Tasks. In Craig Knoblock and Subbarao Kambhampati, editors, *Proceedings of IJCAI-03 Workshop on Information Integration*, pages 73–78, Acapulco, Mexico, August 2003.

[26] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, December 1999.

[27] Elon S. Correa, Alex A. Freitas, and Colin G. Johnson. A new discrete particle swarm algorithm applied to attribute selection in a bioinformatics data set. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 35–42, New York, NY, USA, 2006. ACM.

[28] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. AgreementMaker: efficient matching for large real-world schemas and ontologies. *Proc. VLDB Endow.*, 2:1586–1589, August 2009.

[29] Isabel F. Cruz and William Sunna. Structural Alignment Methods with Applications to Geospatial Ontologies. *T. GIS*, 12(6):683–711, 2008.

[30] Mariana Damova, Atanas Kiryakov, Kiril Simov, and Svetoslav Petrov. Mapping the central lod ontologies to proton upper-level ontology. In *Ontology Mapping Workshop at ISWC 2010*, Shanghai, China, 2010.

[31] Jérôme David, Jérôme Euzenat, François Scharffe, and Cössia Trojahn dos Santos. The Alignment API 4.0. In *Semantic Web*, volume 2, pages 3–10, 2010.

[32] Jérôme David, Fabrice Guillet, and Henri Briand. Association Rule Ontology Matching Approach. *International Journal of Semantic Web Information Systems*, 3(2):27–49, 2007.

[33] Jérôme David, Fabrice Guillet, Régis Gras, and Henri Briand. Conceptual hierarchies matching: an approach based on discovery of implication rules between concepts. In *ECAI'06*, pages 357–361, Amsterdam, The Netherlands, The Netherlands, 2006. IOS Press.

[34] Lee R. Dice. Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):pp. 297–302, 1945.

[35] Anhai Doan, Jayant Madhavan, Pedro Domingos, and Alon Halevy. Ontology Matching: A Machine Learning Approach. In *Handbook on Ontologies in Information Systems*, pages 397–416, 2003.

[36] Marc Ehrig. Y.: Foam - framework for ontology alignment and mapping; results of the ontology alignment initiative. In *Proceedings of the Workshop on Integrating Ontologies. Volume 156., CEUR-WS.org (2005) 72?76*, pages 72–76, 2005.

[37] Marc Ehrig. *Ontology Alignment: Bridging the Semantic Gap*, volume 4 of *Semantic Web And Beyond Computing for Human Experience*. Springer, 2007.

[38] Jérôme Euzenat. An API for Ontology Alignment. In *The Semantic Web. ISWC 2004*, pages 698–712. Springer, 2004.

[39] Jérôme Euzenat, Alfio Ferrara, Christian Meilicke, Andriy Nikolov, Juan Pane, François Scharffe, Pavel Shvaiko, Heiner Stuckenschmidt, Ondrej Svüb-Zamazal, Vojtech Svütek, and Cössia Trojahn dos Santos. Results of the ontology alignment evaluation initiative 2010. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel Cruz, editors, *Proc. 5th ISWC workshop on ontology matching (OM), Shanghai (CN)*, pages 85–117, 2010.

[40] Jérôme Euzenat, Christian Meilicke, Pavel Shvaiko, Heiner Stuckenschmidt, and Cássia Trojahn dos Santos. Ontology Alignment Evaluation Initiative: six years of experience. In *Journal on data semantics*, Lecture Notes in Computer Science, chapter XV(6720), pages 158–192. 2011.

[41] Jérôme Euzenat and Pavel Shvaiko. *Ontology Matching*. Springer, 2007.

[42] Jérôme Euzenat and P. Valtchev. Similarity-based ontology alignment in OWL-lite. In R. López de Mántaras and L. Saitta, editors, *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI-04)*, pages 333–337. IOS Press, 2004.

[43] Ronen Feldman and James Sanger. *The Text Mining Handbook: Advanced Approaches in Analysing Unstructured Data*. Cambridge University Press, December 2006.

[44] Christiane Fellbaum, editor. *WordNet An Electronic Lexical Database*. The MIT Press, Cambridge, London, May 1998.

[45] Ivan P. Fellegi and Alan B. Sunter. A Theory for Record Linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[46] John Rupert Firth. A synopsis of linguistic theory 1930–1955. In *Studies in Linguistic Analysis*, pages 1–32. Blackwell, Oxford, 1957.

[47] Avigdor Gal. Why is schema matching tough and what can we do about it? *SIGMOD Rec.*, 35:2–5, December 2006.

[48] Fausto Giunchiglia, Pavel Shvaiko, and Mikalai Yatskevich. S-match: an algorithm and an implementation of semantic matching. In Christoph Bussler, John Davies, Dieter Fensel, and Rudi Studer, editors, *The Semantic Web: Research and Applications*, volume 3053 of *Lecture Notes in Computer Science*, pages 61–75. Springer, 2004.

[49] Fausto Giunchiglia, Mikalai Yatskevich, Paolo Avesani, and Pavel Shivaiko. A large dataset for the evaluation of ontology matching. *The Knowledge Engineering Review*, 24(02):137, May 2009.

[50] O. Gotoh. An improved algorithm for matching biological sequences. *Journal of molecular biology*, 162(3):705–708, December 1982.

[51] Michael Granitzer, Vedran Sabol, Kow Weng Onn, Dickson Lukose, and Klaus Tochtermann. Ontology alignment: A survey with focus on visually supported semi-automatic techniques. *Future Internet*, 2(3):238–258, 2010.

[52] Gregory Grefenstette. Corpus-derived first-, second- and third-order word affinities. In *Euralex*, pages 279–290, 1994.

[53] Hong hai Do and Erhard Rahm. Coma - a system for flexible combination of schema matching approaches. In *In VLDB*, pages 610–621, 2002.

[54] M.A.K. Halliday and J. Webster. *On language and linguistics*. The Collected Works of M. A. K. Halliday. Continuum, 2006.

[55] R. W. Hamming. Error Detecting and Error Correcting Codes. *Bell System Techincal Journal*, 29:147–160, 1950.

[56] Graeme Hirst and David St-Onge. *Lexical Chains as representation of context for the detection and correction malapropisms*. The MIT Press, 1998.

[57] Ian Horrocks, Peter F. Patel-Schneider, and Frank Van Harmelen. From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1:2003, 2003.

[58] Wei Hu and Yuzhong Qu. Falcon-ao: A practical ontology matching system. *Web Semant.*, 6:237–239, September 2008.

[59] Yifan Hu. Efficient, high-quality force-directed graph drawing. *Mathematica Journal*, 10(1):37–71, 2005.

[60] P. Jaccard. Nouvelles recherches sur la distribution florale. *Bulletin de la Sociète Vaudense des Sciences Naturelles*, 44:223–270, 1908.

[61] Prateek Jain, Pascal Hitzler, Amit P. Sheth, Kunal Verma, and Peter Z. Yeh. Ontology Alignment for Linked Open Data. *Information Retrieval*, 9(November):402–417, 2010.

[62] Prateek Jain, Peter Z. Yeh, Kunal Verma, Reymonrod G. Vasquez, Mariana Damova, Pascal Hitzler, and Amit P. Sheth. Contextual ontology alignment of lod with an upper ontology: a case study with proton. In *Proceedings of the 8th extended semantic web conference on The semantic web: research and applications - Volume Part I*, ESWC'11, pages 80–92, Berlin, Heidelberg, 2011. Springer-Verlag.

[63] M. A. Jaro. Probabilistic Linkage of Large Public Health Data Files. *Statistics in Medicine*, 14:491–498, 1995.

[64] Yves R. Jean-Mary, E. Patrick Shironoshita, and Mansur R. Kabuka. Ontology matching with semantic verification. *Web Semantics*, 7:235–251, September 2009.

[65] M. Jenkins and D. Smith. Conservative stemming for search and indexing. *SIGIR'05*, 2005.

[66] Qiu Ji, Peter Haase, and Guilin Qi. Combination of Similarity Measures in Ontology Matching using the OWA Operator. 2008.

[67] Ningsheng Jian, Wei Hu, Gong Cheng, and Yuzhong Qu. Falcon-ao: Aligning ontologies with falcon. In *In: K-Cap 2005 Workshop on Integrating Ontologies*, pages 87–93, 2005.

[68] John McCarthy. Generality In Artificial Intelligence. *Communications of the ACM*, 30:1030–1035, 1987.

[69] John Mccarthy. Notes on Formalizing Context. pages 555–560. Morgan Kaufmann, 1993.

[70] John McCarthy and Sasa Buvac and Tom Costello and Richard Fikes and Mike Genesereth and Fausto Giunchiglia. Formalizing Context (Expanded Notes), 1995.

[71] D. Jones, T. Bench-Capon, and P. Visser. Methodologies for ontology development. In *IT&KNOWS Conference, XV IFIP World Computer Congress*, Budapest, August 1998.

[72] Karen Sparck Jones and Peter Willet. Readings in information retrieval. San Francisco, 1997. Morgan Kaufmann.

[73] Jaewoo Kang and Jeffrey F. Naughton. On schema matching with opaque column names and data values. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, SIGMOD '03, pages 205–216, New York, NY, USA, 2003. ACM.

[74] Anne Kao and Steve R. Poteet. *Natural Language Processing and Text Mining*. Springer, 2007.

[75] Vipul Kashyap and Amit Sheth. Semantic and schematic similarities between database objects: A context-based approach. *VLDB Journal*, 5:276–304, 1996.

[76] J.C. Kendall. *The man who made lists: love, death, madness, and the creation of Roget's Thesaurus*. G.P. Putnam's Sons, 2008.

[77] David Kensche, Christoph Quix, Mohamed Chatti, and Matthias Jarke. GeRoMe: A Generic Role Based Metamodel for Model Management. In Stefano Spaccapietra, Paolo Atzeni, Françis Fages, Mohand-Saïd Hacid, Michael Kifer, John Mylopoulos, Barbara Pernici, Pavel Shvaiko, Juan Trujillo, and Ilya Zaihrayeu, editors, *Journal on Data Semantics VIII*, volume 4380 of *Lecture Notes in Computer Science*, pages 82–117. Springer Berlin / Heidelberg, 2007.

[78] David Kensche, Christoph Quix, Xiang Li, and Yong Li. Geromesuite: a system for holistic generic model management. In *Proceedings of the 33rd international conference on Very large data bases*, VLDB'07, pages 1322–1325. VLDB Endowment, 2007.

[79] Jaehong Kim, Minsu Jang, Young-Guk Ha, Joo-Chan Sohn, and Sang Lee. Moa: Owl ontology merging and alignment tool for the semantic web. In Moonis Ali and Floriana Esposito, editors, *Innovations in Applied Artificial Intelligence*, volume 3533 of *Lecture Notes in Computer Science*, pages 116–123. Springer Berlin / Heidelberg, 2005.

[80] Su N. Kim and Timothy Baldwin. Disambiguating noun compounds. In *AAAI'07: Proceedings of the 22nd national conference on Artificial intelligence*, pages 901–906. AAAI Press, 2007.

[81] Su Nam Kim and Timothy Baldwin. Automatic interpretation of noun compounds using wordnet similarity. In *In Proceedings of the 2nd International Joint Conference on Natural Language Processing, Jeju Island, South Korea*, pages 945–956, 2005.

[82] Su Nam Kim, Meladel Mistica, and Timothy Baldwin. Extending sense collocations in interpreting noun compounds. pages 49–56, 2007.

[83] Patrick Lambrix and A. Edberg. Evaluation of ontology merging tools in bioinformatics. *Pac Symp Biocomput*, pages 589–600, 2003.

[84] Patrick Lambrix and He Tan. Sambo-a system for aligning and merging biomedical ontologies. *Web Semantics, Special issue on Semantic Web for the Life Sciences*, 4:196–206, September 2006.

[85] Patrick Lambrix and He Tan. Journal on data semantics viii. chapter A tool for evaluating ontology alignment strategies, pages 182–202. 2007.

[86] Claudia Leacock and Martin Chodorow. Combining Local Context and WordNet Similarity for Word Sense Identification. *An Electronic Lexical Database*, pages 265–283, 1998.

[87] Lillian Lee. On the Effectiveness of the Skew Divergence for Statistical Language Analysis. In *Artificial Intelligence and Statistics 2001*, pages 65–72, 2001.

[88] G.N. Leech. *Semantics: the study of meaning*. Language and linguistics. Penguin, 1981.

[89] Isaac Lera, Carlos Juiz, and Ramon Puigjaner. Quick ontology mapping algorithm for distributed enrionments. In Hamid R. Arabnia, editor, *Semantic Web and Web Services*, volume 1, pages 107–113. CSREA Press, 2008.

[90] Isaac Lera, Carlos Juiz, and Ramon Puigjaner. Owl-m extension for semantic representations of ontology alignments. In IEEE Computer Society, editor, *2010 International Conference on Complex, Intelligent and Software Intensive System (CISIS). 3rd International Workshop on Ontology Alignment and Visualization - OnAV'10.*, pages 956–961, Krakow, Poland, February, 15-18 2010.

[91] Isaac Lera, Carlos Juiz, and Ramon Puigjaner. Unsupervised algorithm for the concept disambiguation in ontologies - semantic rules and voting system to determine suitable senses. In SciTePress, editor, *International Conference on Knowledge Engineering and Ontology Development (KEOD 2010)*, pages 388–391, Valencia, Spain, October, 25-28 2010.

[92] Vladimir Levenshtein. Binary Codes Capable of Correcting Deletions and Insertions and Reversals. *Soviet Physics Doklady*, 10(8):707–710, 1966.

[93] Juanzi Li, Jie Tang, Yi Li, and Qiong Luo. Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Transactions on Knowledge and Data Engineering*, 21:1218–1232, 2009.

[94] Dekang Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pages 296–304, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[95] Shuang Liu, Clement Yu, and Weiyi Meng. Word sense disambiguation in queries. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, CIKM '05, pages 525–532, New York, NY, USA, 2005. ACM.

[96] L. Lovász and M.D. Plummer. *Matching theory*. AMS Chelsea Publishing Series. AMS Chelsea Pub., 2009.

[97] Jayant Madhavan, Philip A. Bernstein, and Erhard Rahm. Generic Schema Matching with Cupid. In *Proceedings of the 27th International Conference on Very Large Data Bases*, VLDB '01, pages 49–58, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[98] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[99] Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1 edition, June 1999.

[100] Ming Mao and Yefei Peng. The PRIOR+: Results for OAEI Campaign 2007. In Shvaiko et al. [142].

[101] Ming Mao, Yefei Peng, and Michael Spring. A Profile Propagation and Information Retrieval Based Ontology Mapping Approach. In *Proceedings of the Third International Conference on Semantics, Knowledge and Grid*, pages 164–169, Washington, DC, USA, 2007. IEEE Computer Society.

[102] Victor M. Markowitz and Thodoros Topaloglou. Applying Data Warehouse Concepts to Gene Expression Data Management. In *Proceedings of the 2nd IEEE International Symposium on Bioinformatics and Bioengineering*, BIBE '01, pages 65–, Washington, DC, USA, 2001. IEEE Computer Society.

[103] Scott Mcdonald and Michael Ramscar. Testing the distributional hypothesis: The influence of context on judgements of semantic similarity. In *In Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, pages 611–616, 2001.

[104] Scott Mcginnis and Thomas L. Madden. BLAST: at the core of a powerful and diverse set of sequence analysis tools. *Nucleic Acids Res*, 32:20–25, 2004.

[105] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder. An Environment for Merging and Testing Large Ontologies. In *Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000)*, Breckenridge, Colorado, April 2000.

[106] Sergey Melnik, Hector Garcia-Molina, and Erhard Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and Its Application to Schema Matching. In *Proceedings of the 18th International Conference on Data Engineering*, ICDE '02, pages 117–, Washington, DC, USA, 2002. IEEE Computer Society.

[107] Rada Mihalcea and Dan Moldovan. Automatic Generation of a Coarse Grained WordNet. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, New Brunswich, NJ, 2001. Pittsburgh, PA, ACL.

[108] George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. WordNet: A on-line Lexical database. *International Journal of Lexicography*, pages 235–244, 1990.

[109] Renee J. Miller, Mauricio A. Hernandez, Laura M. Haas, Lingling Yan, Howard C. T. Ho, Ronald Fagin, and Lucian Popa. The Clio project: managing heterogeneity. *SIGMOD Rec.*, 30(1):78–83, March 2001.

[110] Marie-Francine Moens. *Information Extraction: Algorithms and Prospects in a Retrieval Context (The Information Retrieval Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[111] Alvaro E. Monge and Charles Elkan. The field matching problem: Algorithms and applications. In *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 267–270, 1996.

[112] Miklos Nagy, Maria Vargas-Vera, and Enrico Motta. Dssim - managing uncertainty on the semantic web. In Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors, *OM*, volume 304 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

[113] Felix Naumann and Melanie Herschel. *An Introduction to Duplicate Detection*. Morgan and Claypool Publishers, 2010.

[114] Roberto Navigli. Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 2009.

[115] Saul Needleman and Christian Wunsch. A general method applicable to the search for similarities in the animo acid sequence of two proteins. *Journal of Molecular Biology*, 3(48):443–453, 1970.

[116] Bruce E. Nevin and Stephen M. Johnson, editors. *The Legacy of Zellig Harris: Language and Information into the 21st Century: Computability of Language and Computer Applications*, volume 2. 2002.

[117] Jan Noessner and Mathias Niepert. Codi: Combinatorial optimization for data integration: results for oaei 2010. In Shvaiko et al. [143].

[118] Natalya F. Noy and Mark A. Musen. The PROMPT suite: interactive tools for ontology merging and mapping. *Int. J. Hum.-Comput. Stud.*, 59:983–1024, December 2003.

[119] Natalya Fridman Noy and Mark A. Musen. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 450–455. AAAI Press, 2000.

[120] Chris D. Paice. Method for evaluation of stemming algorithms based on error counting. *Journal of the American Society for Information Science*, 47(8):632–649, 1996.

[121] Nikhil Pal and Lakhmi C Jain. *Advanced Techniques in Knowledge Discovery and Data Mining (Advanced Information and Knowledge Processing)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[122] Chintan Patel, Kaustubh Supekar, and Yugyung Lee. Ontogenie: Extracting ontology instances from www. In *In Human Language Technology for the Semantic Web and Web Services, ISWC?03*, 2003.

[123] Mikalai Yatskevich Pavel Shvaiko, Fausto Giunchiglia. *Semantic Matching with S-Match*, volume Part 2, pages 183–202. 2010.

[124] M. F. Porter. An algorithm for suffix stripping. *Program*, 3(14):130–137, October 1980.

[125] Raúl Quesada. Identidad y substitución. *Anuario de Filosofía*, 1:17–32, 2007.

[126] Erhard Rahm and Philip A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10:334–350, December 2001.

[127] Sudha Ram and Jinsoo Park. Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema-Level Semantic Conflicts. *IEEE Transactions on Knowledge and Data Engineering*, 16:189–202, 2004.

[128] Sudha Ram, Jinsoo Park, and Dongwon Lee. Digital Libraries for the Next Millennium: Challenges and Research Directions. *Information Systems Frontiers*, 1:75–94, 1999.

[129] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th international joint conference on Artificial intelligence*, volume 1, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[130] Philip Resnik and Mona Diab. Measuring Verb Similarity. In *Proceedings of the 22nd Annual Conference of the Cognitive Science Society*, pages 399–404, 2000.

[131] Philip Resnik and Jimmy Lin. Evaluation of NLP Systems. *Handbook of Computational Linguistics and Natural Language Processing*, pages 271–297, 2010.

[132] Barbara Rosario and Marti Hearst. Classifying the semantic relations in noun compounds via a domain-specific lexical hierarchy. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing (EMNLP-01*, pages 82–90, 2001.

[133] Herbert Rubenstein and John B. Goodenough. Contextual correlates of synonymy. *Commun. ACM*, 8:627–633, October 1965.

[134] Magnus Sahlgren. *The Word-Space Model Using distributional analysis to represent syntagmatic and paradigmatic relations between words*. PhD thesis, Department of Linguistics, Stockholm University, 2006.

[135] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, November 1975.

[136] Hinrich Schötze and Jan O. Pedersen. Information retrieval based on word senses. In *In Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, 1995.

[137] Glenn Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, Princeton, 1976.

[138] Amit P. Sheth. Semantic issues in multidatabase systems - preface by the special issue editor. *SIGMOD Record*, 20(4):5–9, 1991.

[139] Amit P. Sheth and Vipul Kashyap. So Far (Schematically) yet So Near (Semantically). In *Proceedings of the IFIP WG 2.6 Database Semantics Conference on Interoperable Database Systems (DS-5)*, pages 283–312, Amsterdam, The Netherlands, The Netherlands, 1993. North-Holland Publishing Co.

[140] Pavel Shvaiko and Jérôme Euzenat. A Survey of Schema-Based Matching Approaches. In *Journal on Data Semantics IV*, Lecture Notes in Computer Science, chapter 5, pages 146–171. 2005.

[141] Pavel Shvaiko and Jérôme Euzenat. Ten Challenges for Ontology Matching. In Robert Meersman and Zahir Tari, editors, *On the Move to Meaningful Internet Systems: OTM 2008*, volume 5332 of *Lecture Notes in Computer Science*, pages 1164–1182. Springer Berlin / Heidelberg, 2008.

[142] Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, and Bin He, editors. *Proceedings of the 2nd International Workshop on Ontology Matching (OM-2007) Collocated with the 6th International Semantic Web Conference (ISWC-2007) and the 2nd Asian Semantic Web Conference (ASWC-2007), Busan, Korea, November 11, 2007*, volume 304 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2008.

[143] Pavel Shvaiko, Jérôme Euzenat, Fausto Giunchiglia, Heiner Stuckenschmidt, Ming Mao, and Isabel F. Cruz, editors. *Proceedings of the 5th International Workshop on Ontology Matching (OM-2010), Shanghai, China, November 7, 2010*, volume 689 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2010.

[144] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981.

[145] Vassilis Spiliopoulos, Alexandros G. Valarakos, George A. Vouros, and Vangelis Karkaletsis. SEMA: Results for the Ontology Alignment Contest OAEI 2007. In Shvaiko et al. [142].

[146] Jason Stanley. Context and logical form. *Linguistics and Philosophy*, 23(4):391–434, 2000.

[147] Mark Stevenson and Yorick Wilks. The interaction of knowledge sources in word sense disambiguation. *Comput. Linguist.*, 27(3):321–349, 2001.

[148] Giorgos Stoilos, Giorgos Stamou, and Stefanos Kollias. A String Metric for Ontology Alignment. pages 624–637, 2005.

[149] Thomas Strang and Claudia Linnhoff-Popien. A context modeling survey. In *In: Workshop on Advanced Context Modelling, Reasoning and Management, UbiComp 2004 - The Sixth International Conference on Ubiquitous Computing, Nottingham/England*, 2004.

[150] Thomas Strang, Claudia Linnhoff-Popien, and Korbinian Frank. Applications of a context ontology language. In *University of Split, Croatia*, pages 14–18, 2003.

[151] G. Stumme and A. Maedche. FCA–Merge: Bottom-Up Merging of Ontologies. In *IJCAI-2001 – Proceedings of the 17th International Joint Conference on Artificial Intelligence, Seattle, USA, August, 1-6, 2001*, pages 225–234, San Francisco, 2001. Morgen Kaufmann.

[152] Michael John Sussna. *Text retrieval using inference in semantic metanetworks*. PhD thesis, La Jolla, CA, USA, 1997. UMI Order No. GAX97-26031.

[153] Alexandros G. Valarakos, Ros G. Valarakos, Georgios Paliouras, Vangelis Karkaletsis, and George Vouros. A name-matching algorithm for supporting ontology enrichment. In *In Proceedings of SETN?04, 3rd Hellenic Conference on Artificial Intelligence*, pages 381–389. Springer Verlag, 2004.

[154] C.J. van Rijsbergen, S.E. Robertson, and M.F. Porter. *New models in probabilistic information retrieval*. London: British Library. (British Library Research and Development Report, no. 5587), 1980.

[155] W3C. Linked Data. `http://www.w3.org/standards/semanticweb/data`, 2010.

[156] W3C Recommendation. Architecture of the World Wide Web, Volume One. `http://www.w3.org/TR/2004/REC-webarch-20041215/`, 2004.

[157] W3C Recommendation. RDF Primer. `http://www.w3.org/TR/rdf-syntax/`, 2004.

[158] W3C Recommendation. Extensible Markup Language (XML) 1.0 (Fifth Edition). `http://www.w3.org/TR/xml/`, 2008.

[159] Peng Wang and Baowen Xu. LILY: the Results for the Ontology Alignment Contest OAEI 2007. In Shvaiko et al. [142].

[160] Julie Elizabeth Weeds. *Measures and Applications of Lexical Distributional Similarity*. PhD thesis, 2003.

[161] Mark Weiser. Hot topics: Ubiquitous computing. *IEEE Computer*, 26(10):71–72, October 1993.

[162] Sholom Weiss, Nitin Indurkhya, Tong Zhang, and Fred Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*. SpringerVerlag, 2004.

[163] William E. Winkler. The state of record linkage and current research problems. Technical report, Statistical Research Division, U.S. Bureau of the Census, 1999.

[164] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, ACL '94, pages 133–138, Stroudsburg, PA, USA, 1994. Association for Computational Linguistics.

[165] Dongqiang Yang and David M. W. Powers. Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science - Volume 38*, ACSC '05, pages 315–322, Darlinghurst, Australia, Australia, 2005. Australian Computer Society, Inc.

[166] Xia Yang, Mong Lee, and Tok Ling. Resolving structural conflicts in the integration of xml schemas: A semantic approach. In Il-Yeol Song, Stephen Liddle, Tok-Wang Ling, and Peter Scheuermann, editors, *Conceptual Modeling - ER 2003*, volume 2813 of *Lecture Notes in Computer Science*, pages 520–533. Springer Berlin / Heidelberg, 2003.

[167] J. Zobel and P. W. Dart. Phonetic String Matching: Lessons from Information Retrieval. In H. P. Frei, D. Harman, P. Schäble, and R. Wilkinson, editors, *Proceedings of the 19th International Conference on Research and Development in Information Retrieval*, pages 166–172, Zurich, Switzerland, 1996. ACM Press.