Lengoaia eta Sistema Informatikoak Saila
Departamento de Lenguajes y Sistemas Informáticos

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

LiDom Builder: Automatising the Construction of Multilingual Domain Modules

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy by

Ángel Conde Manjón

January, 2016

Lengoaia eta Sistema Informatikoak Saila
Departamento de Lenguajes y Sistemas Informáticos

eman ta zabal zazu

Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

LiDom Builder: Automatising the Construction of Multilingual Domain Modules

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy by Ángel Conde Manjón under the supervision of Dr. Ana Arruarte Lasa and Dr. Mikel Larrañaga Olagaray

January, 2016

Acknowledgements

After a considerable effort, it is time to thank all the people who made this work feasible.

Mikel eta Anari, hasieratik eta azken momentu arte eskainitako laguntzagatik.

Mila esker Joniri, nigan beti adierazi duen konfidantzagatik.

I want to thank all the members of the GaLan research group for their support. Specifically, I would like thank Isabel Fernández de Castro for her work coordinating the different research interests of the group.

Eneko, Aitor eta Oierri Eskerrik asko zuen laguntzagatik.

I would also like to thank the Illinois Cognitive Computation Group and, especially, Dan Roth for his invaluable comments.

A los compañeros del departamento en Vitoria, que me habéis aceptado como a uno más después de haber sido vuestro alumno.

A mis amigos, que siempre están para apoyarme.

A mis padres y hermana, por ese ánimo y apoyo incansable a que me siga formando.

And last but not least ... a Amaia, por haber estado a mi lado durante este tiempo.

Abstract

Nowadays, in the developed world, it is almost impossible to deal with any situation without the use of Information and Communication Technologies, and Education is no exception to this. In the area of Technology Supported Learning Systems, for example, new means and mechanisms that take advantage of their development and use must provide solutions to problems such as bilingual or multilingual education. It would be appreciated if a Technology Supported Learning System could deal with the management of a subject in more than one language.

This thesis presents *LiDom Builder*, a framework for the automatic generation of *Multilingual Domain Modules* from electronic documents. In *LiDom Builder*, the domain module representation relies on an ontology that represents the topics to be mastered along with the pedagogical relationships between them, and the set of Didactic Resources to be used in the learning sessions. The formalism provided for representing the domain is valid to deal with multilingual domains as it allows both the topics of the domain and the didactic resources to be used throughout the learning sessions in every supported language to be represented.

The *LiDom Builder* framework consists of three main modules that perform the acquisition of the different elements that constitute the *Domain Module*: topics of the domain, pedagogical relationships between those topics, and Learning Objects or Didactic Resources annotated with metadata. *LiDom Builder* uses techniques such as Ontology Learning or Machine Learning, along with resources such as Wikipedia to fulfil its work.

Contents

List of Figures

List of Tables

List of Listings

List of Algorithms

1 Introduction

This chapter contains the motivation, reasons and goals behind this thesis. It remarks the benefits of bilingual and multilingual education in Technology Supported Learning Systems (TSLSs), and the necessity that currently exists not only for building multilingual learning systems, but also for automatising the acquisition process of core components such as the *Domain Module.* The followed working methodology along with the contextualization of the work inside the GaLan research group are presented before finishing with the outline of the work of this thesis.

1.1 Motivation and Goals

The term Education refers to the systematic and voluntary activities aimed at satisfying lifelong learning necessities, either in formal or informal contexts, with the intention of encouraging equal opportunities, social cohesion and active citizenship. Language and, in particular, the choice of language of instruction is a big concern in the current context of *Education for all.* The language of instruction is the medium of communication for the transmission of knowledge. Bilingual, and multilingual contexts in Education are nowadays a reality. The challenge is for education systems to adapt to these complex realities and provide a quality education (Unesco, 2003).

In technologically developed societies, this reality of bilingual and multilingual education has a strong influence in Information and Communication Technologies (ICT) in general, and in Technology Supported Learning Systems (TSLSs)in particular. Years of research have facilitated the development of different kinds of TSLSs such as Learning Management Systems, Intelligent Tutoring Systems (ITSs), Collaborative Learning Systems, or Web-based Educational Systems. It would be useful if a TSLS could deal with the management of a subject in more than one language. This would benefit both the main communities implied in a teaching/learning process. On the one hand, students would have the opportunity to learn a subject in the language they choose. On the other hand, teachers would have the chance to teach in the language they want. As regards TSLSs, this would necessarily imply the capacity to deal with more than one language, the capacity to represent the domain knowledge from a multilingual point of view.

In order to be effective, any kind of TSLS requires an appropriate *Domain Module*, i.e., the pedagogical representation of the domain to be mastered. The *Domain Module* contains the ideal expert knowledge and also the bugs, mal-rules, and misconceptions that students periodically exhibit (Anderson, 1988; Nkambou et al., 2010; Wenger, 1987; Woolf, 2008). The *Domain Module* enables either the students to learn by themselves, in the case of exploratory learning systems, or the TSLSs to plan the learning process in instructivist systems. But even in a time when TSLSs are being widely used in formal or informal educational scenarios, there is a lack of tools that allow their development in an automatic or semi-automatic way.

Content authoring is a time and effort consuming task. Therefore, efforts in automatising the *Domain Module* acquisition are necessary (Murray, 1999). The construction of *Domain Modules* is a hard task that might become easier by reusing existing materials (Casey and McAlpine, 2003). *Domain Modules* authoring entails not only selecting the domain topics to be learnt, but also defining the pedagogical relationships among the topics, content sequencing, etc. In addition, the proliferation of Learning Objects (LOs), i.e. reusable resources with educational purposes, and Learning Object Repositories (LORs) might help to reduce the development cost of the learning material to be used (Downes, 2003).

The automatic or semiautomatic generation of the *Domain Module* for TSLSs from electronic documents has been rarely addressed. KONGZI (Lu et al., 1995) is a system for automatically building ITSs from machine readable representations of textbooks. The system requires the instructional designers to transcribe the textbook to a machine readable format using a formal descriptive language. Lentini et al. (2000) proposed an environment to build ITSs from spreadsheets in the domain of mathematics. Some other approaches, IMAT (Hoog et al., 1999) and ALOCOM (Verbert et al., 2008; Verbert, 2008), focus on the disaggregation and the reuse of electronic documents for building new learning material. The Knowledge Puzzle project (Zouaq and Nkambou, 2009) was developed to improve Learning Objects (LOs) with instructional and domain knowledge, which is gathered from the LOs using Natural Language Processing (NLP) techniques. Arikiturri (Aldabe, 2011; Aldabe and Maritxalar, 2014) is a tool for building exercises from text corpora based on NLP techniques. Both KONGZI and the system of Lentini et al. are restricted to a particular kind of TSLSs and do not allow interoperability with other systems. The other described systems are standard and specification compliant, but do not support the authoring of the whole *Domain Module*.

DOM-Sortze (Larrañaga, 2012; Larrañaga et al., 2014) is a suite of applications and web-services aimed at the semiautomatic development of *Domain Modules* from electronic textbooks. It is a domain independent tool that combines NLP techniques with heuristic reasoning and ontologies. Although is intended to be able to deal with different languages, it was initially applied to textbooks written in the Basque language.

The final goal of our work is the design and development of computer applications aimed at the automatic acquisition of *Multilingual Domain Modules* for TSLSs. *LiDom Builder* will constitute an evolution of *DOM-Sortze* in that direction. The management of more than one input language in the generation of *Domain Modules* from electronic documents along with the use of techniques such as heuristic reasoning, NLP and Machine Learning and additional resources, such as Wikipedia, will be helpful to achieve this goal.

1.2 Working Methodology

This thesis describes *LiDom Builder*, a framework that enables the automatic generation of *Multilingual Domain Modules* for TSLSs from electronic documents. In the context of this thesis, a *Multilingual Domain Module* is intended to represent a certain domain with educational purposes in more than one language. In the transition from *monolingual* to *Multilingual Domain Modules*, *LiDom Builder* can be considered an evolution of *DOM-Sortze*, which incorporates not only new techniques, but also the use of additional resources such as Wikipedia.

The *LiDom Builder* framework consists of several modules that perform the acquisition of the different elements of the *Domain Module*. Its modular design has facilitated carrying out an incremental development of the framework. Widely used Software Engineering methodologies and approaches have been used to incrementally develop a modular, flexible, and multiplatform framework.

Whilst some of the modules of *LiDom Builder* are completely new, others have been reused and improved from *DOM-Sortze*. For every developed new module, the following procedure has been conducted: first, an analysis of the state-of-the-art has been performed to determine the adequate approach and means to deal with its purpose. The module was then implemented and evaluated using a *Gold-standard* approach. A team of instructional designers has defined the expected reference results, which have been compared to those automatically obtained by *LiDom Builder*. The performance of the different modules has been measured in terms of *precision*, i.e., the proportion of extracted elements that are in the *Gold-standard*, and *recall*, i.e., the percentage of elements in the *Gold-standard* extracted by the system. In addition, *F1-Score*, the harmonic mean of *precision* and *recall* has been also measured.

Furthermore, an expert board has also carried out an expert validation of the outcomes. For validating each module in a particular language, it has been necessary to tune it up previously. In this work, documents in the English language have been used as the source of information.

1.3 Context

The work here presented has been developed in the GaLan[1] research group. This research group, located in the University of the Basque Country (UPV/EHU), has been carrying out its research activity in the area of Computer-aided Education since the early 90's. The main activity of the group is focused on the development of architectures and tools for educational purposes. The GaLan group includes a multidisciplinary team whose members belong to two departments of the UPV/EHU ("Computer Languages and Systems" and "Computer Science and Artificial Intelligence"). Their particular backgrounds integrate different aspects relevant to the development of educational tools.

The GaLan research group has worked on the development of authoring tools for TSLSs among other research lines, in particular tools that support the construction of ITSs (Arruarte, 1998; Arruarte et al., 2003), and tools supporting the semi-automatic elicitation of *Domain Modules* from electronic textbooks (Larrañaga, 2012; Larrañaga et al., 2014).

These two research lines are the pillars which sustain the work presented in this dissertation, which aims to go a step further in the construction of *Multilingual Domain Modules* by automatising the construction of *Domain Modules*.

1.4 Outline

This dissertation is divided in 7 chapters.

Chapter 2 describes *DOM-Sortze* and how to move from *DOM-Sortze* to *LiDom Builder*.

[1] http://galan.ehu.eus

Chapter 3 presents a review of the literature describing the trends, techniques and resources used for knowledge extraction, providing the basis of the work presented throughout this thesis.

Chapter 4 presents the module responsible for eliciting the topics that will constitute the domain to be learnt from electronic documents. This will be the module responsible for identifying multilingual terminology.

Chapter 5 addresses the identification of pedagogical relationships from electronic textbooks, extracting them from both the outline of the textbooks and the full text.

Chapter 6 addresses the identification and extraction of multilingual Didactic Resources using as input not only the electronic textbooks, but also Wikipedia.

Finally, some conclusions and future work are presented in Chapter 7.

2 From DOM-Sortze to LiDom Builder

The ultimate goal of *LiDom Builder* is to deal with multilingual domains. This chapter delves into the issues necessary for *DOM-Sortze* to evolve into *LiDom Builder* in order to pursue this objective. To determine whether the approach followed by *DOM-Sortze* (see Section 2.1) is not restricted to a particular language, Basque in this case, and is suitable to be applied to the generation of *Multilingual Domain Modules*, two questions must be addressed:

- Is the formalism used to represent the domaim appropiate for representing multilingual domains? If this is not so, how should the formalism be enhanced to support them? (Section 2.2)
- Can be the procedure and modules be enhanced to be able to deal with different languages? (Section 2.3)

Next, each of these questions is addressed.

2.1 DOM-Sortze

DOM-Sortze (Larrañaga, 2012; Larrañaga et al., 2014) is a suite of applications and web-services aimed at the semiautomatic development of *Domain Modules* from electronic textbooks. Traditionally, textbooks have been used as the main mechanism to maintain and transmit the knowledge of a certain subject or domain. Textbooks have been authored by domain experts who have organised the contents in a manner that facilitates the understanding and learning, taking into account pedagogical issues. Given that textbooks are appropriate sources of information, they can be used to facilitate the development of the *Domain Module*, allowing the identification of the topics to be mastered and the pedagogical relationships among them, as well as the extraction of LOs, i.e. meaningful fragments of the textbook suitable for educational purposes.

DOM-Sortze was designed and developed with the aim of automatising the development of the *Domain Module*, regardless of the subject, promoting knowledge reuse. *DOM-Sortze* uses NLP techniques, heuristic reasoning and ontologies to fulfil its work. The Basque language was chosen for the experimental work and evaluation.

In *DOM-Sortze*, the *Domain Module* encodes knowledge at two different levels: (1) the knowledge to be learnt, including the topics and the pedagogical relationships that enable planning and determining the learning sessions, which is described by the Learning Domain Ontology (LDO), and (2) the set of LOs that will be used for each domain topic. Using an ontology to describe the learning topics and the pedagogical relationships among the topics will facilitate reusing the described *Domain Module* in different TSLSs after the convenient (automatic) ontology mapping or translation (Uschold and Gruninger, 1996). In *DOM-Sortze*, the following steps are carried out to develop the *Domain Module* (see Figure 2.1):

1. **Document preprocessing**: First, the document must be prepared for the subsequent knowledge acquisition processes. The outcomes of the document preprocessing phase are used to gather the two levels of knowledge encoded in the *Domain Module*. The outline of the document is suitable for the construction of the LDO, while the content of the document is useful for both building the LDO and generating LOs.

2. **Gathering the LDO**: At this phase, the domain topics to be mastered, along with the pedagogical relationships among them, are identified and described in the LDO. The LDO ontology can be used in different ways for learning. On the one hand, instructivist TSLSs will use this information to plan the learning sessions. On the other hand, the students can rely on the LDO to guide them during the learning process.

3. **Gathering the LOs**: At this stage the LOs –definitions, examples, exercises, etc.– to be used during the learning process are identified and generated.

2.2 Domain Module Representation Formalism

In the context of this thesis, a *Multilingual Domain Module* is intended to represent a certain domain with educational purposes in more than one language. In

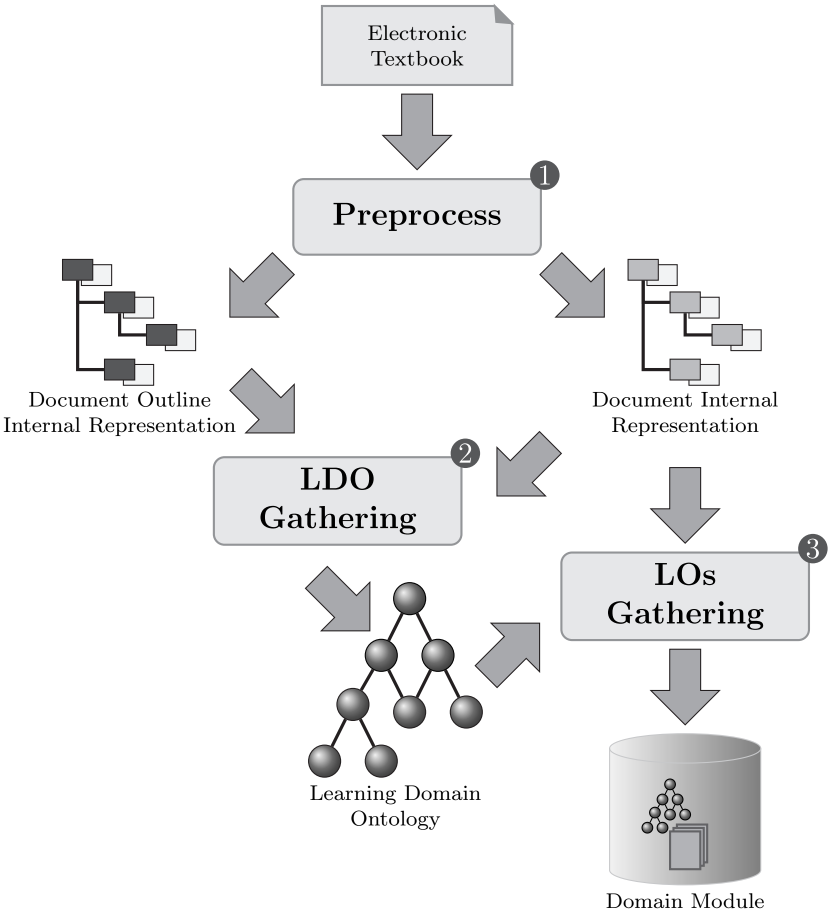


Figure 2.1 – Domain Module Building Process (from Larrañaga, 2012)

DOM-Sortze, the *Domain Module* representation relies on the LDO –an ontology that represents the topics to be mastered along with the pedagogical relationships between them– and the set of LOs to be used in the learning sessions. So far, the LDO has been used to represent the domain in a particular language, Basque in the case of *DOM-Sortze*.

Figure 2.2 shows a fragment of a LDO. For the sake of readability, the example is presented in English. Every topic (*Planetary System*, *Solar System*, *Planet*, *Earth*, *Moon*, *Satellite*) entails a unique title, i.e., the topic descriptor in that language. The topics *Earth* and *Moon* are *partOf* the *Solar System*, i.e., they are lower granularity elements that are constituents of the more general topic *Solar System. Earth* is related to *Planet* by the *isA* relationship; in other words, *Earth* is a particular instance of the *Planet* topic. The *prerequisite* relationship between *Satellite* and *Planet* expresses that the latter should be learnt before attempting to learn *Satellite*. Finally, the *pedagogicallyClose* relationship expressed between *Earth* and *Moon* shows that those topics are strongly related and they could be learnt at the same time.

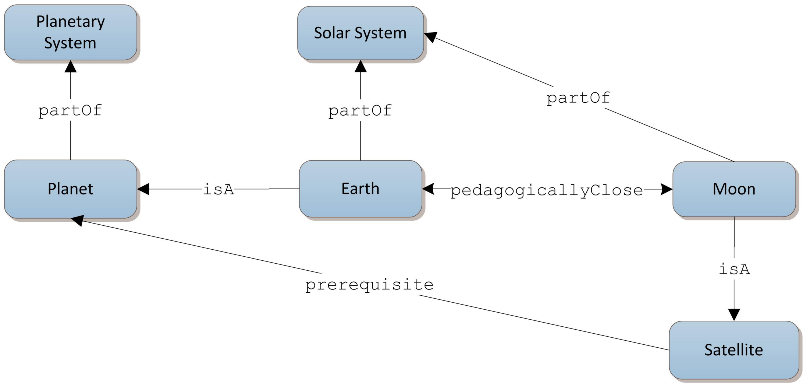


Figure 2.2 – Example of a Learning Domain Ontology Fragment

However, this representation of the domain topics is not enough to represent a multilingual domain. A multilingual representation of the domain requires a multilingual LDO. It should provide the titles of the topics in every supported language. Therefore, the formalism used in *DOM-Sortze* has to be extended to link every topic to its equivalent titles in all the supported languages (see Figure 2.3).

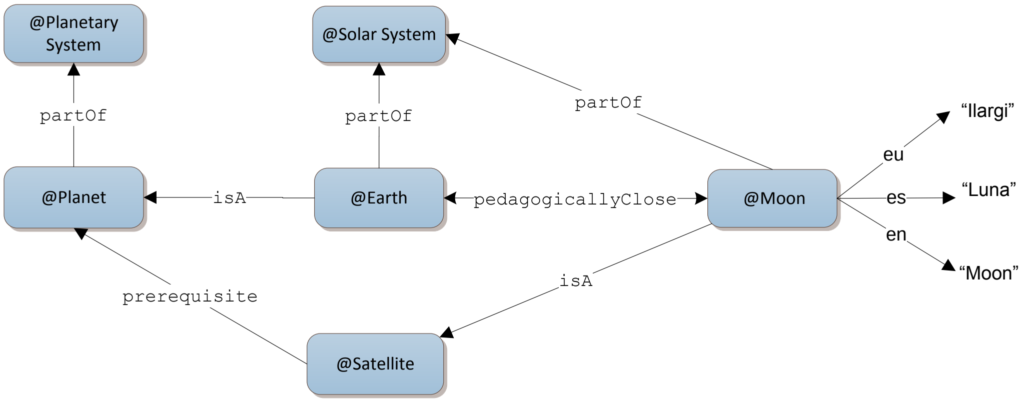


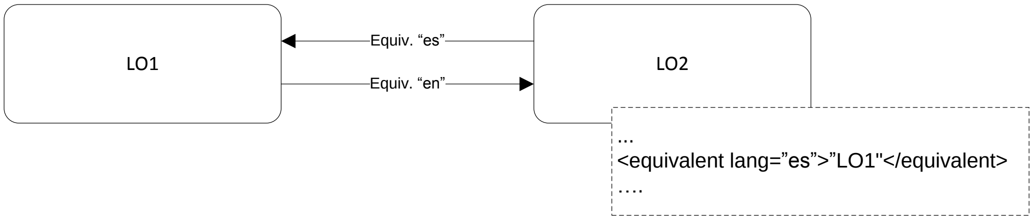
Figure 2.3 – Example of a Multilingual Learning Domain Ontology Fragment

Listing 2.1 shows a fragment of a multilingual LDO represented in OWL.

Listing 2.1 – Example of a Learning Domain Ontology Represented in OWL

```
<!-- http://lsi.vc.ehu.es/Ont#TMoon -->
<LearningDomain:Concept rdf:about="#TMoon">
        <rdfs:label xml:lang="eu">Ilargia</rdfs:label>
        <rdfs:label xml:lang="en">Moon</rdfs:label>
        <rdfs:label xml:lang="es">Luna</rdfs:label>
        <LearningDomain:hasDifficultyLevel
                rdf:resource="&LearningDomain;LOW"/>
        <LearningDomain:hasRelevance
                rdf:resource="&LearningDomain;MEDIUM"/>
        <LearningDomain:isA
                rdf:resource="#TSatellite"/>
        <LearningDomain:pedagogicallyClose
                rdf:resource="#TEarth"/>
</LearningDomain:Concept>
```

In addition to the multilingual LDO, a *Multilingual Domain Module* must also provide the LOs to be used throughout the learning sessions in every supported language. Furthermore, each LOs should be linked to its equivalents in other languages to facilitate their search and retrieval. Therefore, the metadata of each LO has to be improved to describe the link to their equivalents in the other languages (see Figure 2.4).



```
...
<equivalent lang=”es”>”LO1"</equivalent>
….
```

Figure 2.4 – Example of Multilingual Learning Objects

The inclusion of all the above improvements will result in a new formalism suitable for representing *Multilingual Domain Modules* (see Figure 2.5).

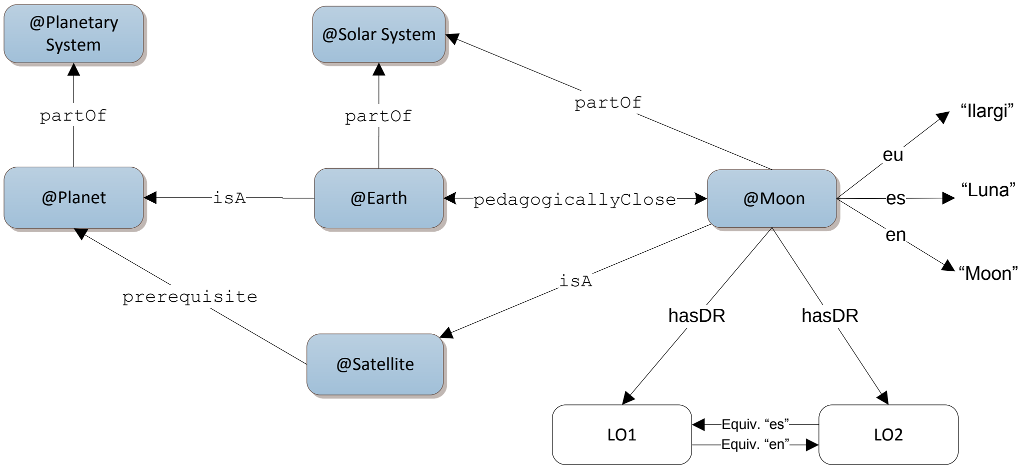


Figure 2.5 – Fragment of a Multilingual Domain Module

2.3 Domain Module Acquisition Process

In this section, the improvements needed in the approach taken by *DOM-Sortze*, so that it can deal with more than one language, are described. To this end, the *Domain Module* acquisition process proposed in *DOM-Sortze* has been analysed and the language-specific resources and tools identified in each step.

1. **Document preprocessing**: First, the document was prepared for the subsequent knowledge acquisition processes. An internal representation of the document, enriched with *part-of-speech* information, was obtained. This internal representation would be used in the following two steps. In the case of Basque, the language in which *DOM-Sortze* was tested, EUSLEM (Aduriz et al., 1996) was the NLP parser used.

2. **Gathering the LDO**: The domain topics to be mastered, along with the pedagogical relationships among them, are identified and described in the LDO. In the case of Basque, *DOM-Sortze* uses Erauzterm (Alegria et al., 2004) for the identification of new topics from the document body, and both a set of heuristics and a grammar to identify pedagogical relationships from the document outline and the document body respectively.

3. **Gathering the LOs**: The LOs –definitions, examples, exercises, etc.– to be used during the learning process are identified and generated. This step is a grammar-based process in which the LDO ontology is used to drive the identification of meaningful fragments related to the domain topics. In the case of

Basque, a grammar that describes the most common syntactic structures used to express the supported kinds of Didactic Resources (DRs) and the discourse markers for Basque are used.

Although the described procedure is generic enough to deal with different languages, in each step some language specific tools are needed (see Figure 2.6). In addition, *LiDom Builder* has firstly to identify the language the document is written in to determine which of those language-dependent resources and tools have to be used.

For the preprocess, a NLP parser for the new supported language must be integrated. Regarding the acquisition of the LDO, it comprises both the elicitation of the multilingual topics to be learnt and the extraction of the pedagogical relationships between those topics. To move from *DOM-Sortze* to *LiDom Builder*, a term extractor that supports the acquisition of multilingual topics must be developed (Chapter 4). The elicitation of pedagogical relationships in *DOM-Sortze* is carried out in two steps, a heuristic-based analysis of the document outline and a grammar-based analysis of the document body. In the transition from *DOM-Sortze* to *LiDom Builder* a new relationship extractor has been developed (Chapter 5). Regarding the extraction of LOs, *LiDom Builder* requires not only new grammars, but also some enhancement to deal with the elicitation of multilingual LOs (Chapter 6).

Although *LiDom Builder* will be designed and developed to be able to deal with different languages, the prototype presented throughout this thesis will work on documents written in English and, thus, will be evaluated on documents written in that language. Therefore, the examples of the techniques presented in this work will be for English.

2.4 Summary

In this section, the main characteristics and limitations of *DOM-Sortze* for dealing with *Multilingual Domain Modules* have been presented. In the transition from *DOM-Sortze* to *LiDom Builder*, i.e., from *Monolingual Domain Modules* to *Multilingual Domain Modules*, some enhancements are needed. These improvements will require new techniques and additional resources. The next section presents a review of these.

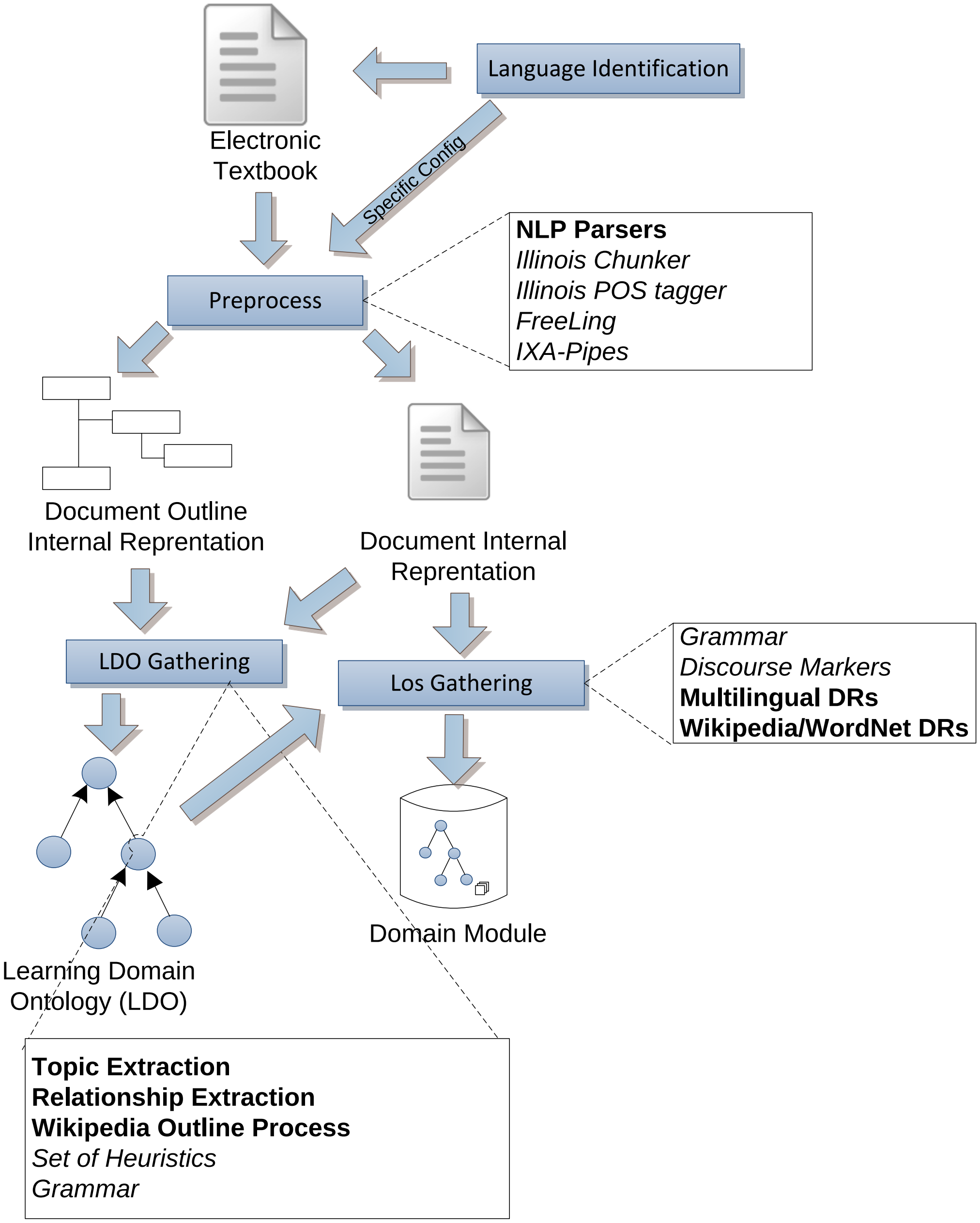


Figure 2.6 – Proposed Multilingual Domain Module Acquisition Process

3 State-of-the-Art

Major efforts have been conducted on Ontology Learning, i.e., semi-automatic processes for the construction of domain ontologies from diverse sources of information. In the last few years, a research trend has focused on the construction of Educational Ontologies, i.e., ontologies aimed at being used for educational purposes.

This section briefly reviews the main characteristics of Educational Ontologies and Learning Ontologies (Section 3.1), some Information Retrieval Techniques for Term Extraction (Section 3.2), and Elicitation of Relationships (Section 3.3). Additional resources used for automatic information elicitation such as Wikipedia, are also presented (Section 3.4).

3.1 Educational Ontologies and Ontology Learning

The ontology term has been adopted from philosophy, where it is defined as the "theory of existence". There are many definitions for ontologies in the area of Computer Science. Neches et al. (1991) proposed the following definition: *"an ontology defines the basic terms and relations comprising the vocabulary of the topic area as well as the rules for combining terms and relations to define extensions to the vocabulary"*. However, Gruber (1991) made the most popular definition of ontologies, which states that *"an ontology is an explicit explanation of a conceptualization"*. This definition was slightly enhanced by Borst (1997), who referred to ontologies as *"formal specifications of a shared conceptualization"*.

According to Studer et al. (1998), "*conceptualization refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon. Explicit means that the type of concepts used, and the constraints on* *their use are explicitly defined. Formal refers to the fact that the ontology should be machine-readable. Shared reflects the notion that an ontology captures consensual knowledge, that is, it is not private to some individual, but accepted by a group*". Ontologies aim at capturing and describing domain knowledge in a generic way and providing a commonly agreed understanding of a domain, which may be reused and shared across applications and groups (Chandrasekaran et al., 1999). They arose as a means to obtain shareable and reusable knowledge bases (Gruber, 1991) and are the core of the Semantic Web (Berners-Lee and Fischetti, 1999; Berners-Lee et al., 2001).

In the days when ontologies have been adopted in many research communities as a way to share, reuse, and process domain knowledge, the TSLSs research community is not an exception. In this community, ontologies present new opportunities, as they provide a great potential by allowing the sharing and reusing of information across learning systems and enabling personalized learner support. The use of ontological engineering, which aims at providing a basis for building models of all things in which computer science is interested (Mizoguchi et al., 1997), was proposed to overcome common problems in the Artificial Intelligence in Education area (Mizoguchi and Bourdeau, 2000). Mizoguchi and Bourdeau (2000) argued that sharing or reusing of knowledge and components could benefit from the use of ontology-based architectures and appropriate ontologies. Dicheva et al. (2005) presented one of the first overviews of ontologies for education. They collect and classify the available information in the field and build the Ontologies for the Education O4E Web Portal. However, what do researchers of educational communities understand by ontology? Although there is no consensus of what it refers to, it can be stated that Educational Ontologies refer to ontologies aimed at being used with educational purposes inside a TSLS. Fok and Ip (2007) define an Educational Ontology as an ontology that can help to retrieve, organise, and recommend educational resources for personalized learning. The idea behind an Educational Ontology is that it can be reused by other learning systems with a wide range of teaching/learning methodologies.

Regarding the use of ontologies in the area of TSLSs, they have been mainly used as a means to represent the *Domain Module* (Cassel et al., 2008; Ganapathi et al., 2011; Jiang et al., 2008; Martin and Mitrovic, 2003; Mitrovic et al., 2003, 2004; Murray, 1998, 2003; Robin and Uma, 2011; Sosnovsky and Gavrilova, 2006), as a mechanism to describe instructional theories (Bourdeau et al., 2004), or to build reusable and scrutable student models (Kay, 1999; Kay and Lum, 2004).

The construction of ontologies and their population, with instantiations of both concepts and relationships, has been commonly called Ontology Learning (Cimiano, Philipp, 2014). Ontology Learning refers to the application of a set of methods and techniques to enable the (semi-)automatic population of ontologies or the construction of ontologies from scratch from diverse information sources (Buitelaar et al., 2005; Pazienza and Stellato, 2012).

Ontology Learning can also contribute to the development of Educational Ontologies, which describe the information about the topics to be mastered along with the pedagogical knowledge (e.g., pedagogical relationships among the topics) required by TSLSs. Gavrilova and colleagues followed a five-step procedure to build an Educational Ontology for C programming (Gavrilova et al., 2005; Sosnovsky and Gavrilova, 2006):

- **Glossary development**: selecting all the essential topics in the domain.
- **Laddering**: structuring the topics of the ontology defining taxonomies, parthood relationships, etc.
- **Disintegration**: Break high level concepts –big concepts– into a set of detailed ones –smaller concepts– where it is needed, using a top-down strategy.
- **Categorization**: Group similar concepts and create meta-concepts to generalize the groups via bottom-up structuring strategy.
- **Refinement**: Update the visual structure by excluding the excessiveness, synonymy, and contradictions.

The same approach was followed to build the Java Learning Object Ontology (JLOO) by Ganapathi et al. (2011) whilst Fok and Ip (2007) took a different approach, reusing existing domain ontologies and adapting them to build the Personalized Education Ontology (PEOnto).

The work presented throughout this thesis focuses on the both the term extraction process and the pedagogical relationships identification process for Educational Ontologies. Therefore, relevant aspects on term extraction, along with some term extraction techniques, and techniques for the extraction of relationships are outlined below.

3.2 Term Extraction

The main goal of term extraction is to identify and extract the most relevant terms in the analysed source of information. This is the first and also one of the essential tasks for Ontology Learning. Term extraction is widely used in text mining and information retrieval, e.g., for indexing scientific literature according to keyphrases and main topics. Term extraction techniques are quite diverse, ranging from linguistic methods, which rely on the detection of the specific syntactic patterns in order to extract the terms, to statistical methods that determine the *termhood* of a candidate term. The *termhood* represents the degree of a linguistic being related to or representing a domain-specific concept (Kageura and Umino, 1996). Finally, the actual trend in term extraction approaches is to combine both kinds of techniques in those so called hybrid approaches. In the following subsections those approaches are described:

3.2.1 Linguistic Approaches

The linguistic approaches for term extraction rely on the syntactic properties of the terms for their identification. These kinds of techniques work under the assumption that terms commonly present characteristic syntactic structures (Benveniste, 1966; Bourigault, 1996) when the *part-of-speech* of those terms is analysed. Furthermore, in a study Daille et al. (1996) confirmed empirically that most terms appear in the form of short noun phrases.

Table 3.1 – Examples of Syntactic Patterns for Term Extraction

Syntactic Pattern	**Examples**
$\mathit{Noun}^{+}\mathit{Noun}$	computer science, solar system, hubble space telescope
$(\mathit{Adj}\vert\mathit{Noun})^{+}\ \mathit{Noun}^{+}$	extra-solar planets, elliptical galaxies, giant tidal waves
$((\mathit{Noun\ Prep}^{?})(\mathit{Adj}\vert\mathit{Noun})^{*}\mathit{Noun}$	coloboma of retina, scotomas in low vision
$((\mathit{Adj}\vert\mathit{Noun})^{+}\vert(\mathit{Adj}\vert\mathit{Noun})^{*}(\mathit{Noun\ Prep})^{?})(\mathit{Adj/Noun})^{*})\mathit{Noun}$	acute exacerbation of chronic bronchitis

Linguistic term extraction approaches apply the following procedure:

1. Perform a shallow linguistic analysis to enrich the analysed text with *part-of-speech* information (e.g., nouns, verbs and adjectives). To fulfill such a task, a *part-of-speech* tagger such as the Stanford Log-Linear Part-Of-Speech Tagger (Toutanova et al., 2003) or FreeLing (Padró and Stanilovsky, 2012) is required.

2. Identify and extract candidate terms through admissible surface forms or shallow parsing grammars (Buitelaar et al., 2005). Table 3.1 provides some examples of the syntactic patterns that are frequently used for term extraction. Some works, e.g., (Daille et al., 1996), also deal with the identification and grouping of meaning-preserving term variants. For instance, the terms *"mission of spacecraft"* and *"spacecraft mission"* refer to the same topic. Therefore, both might be identified as meaning-preserving variants of the same term.

3. Apply linguistic filters, e.g., a list of words (*stopwords*) that will be filtered out to refine the terminology.

3.2.2 Statistical Approaches

Statistical measures provide a means to distinguish among true and false terms given a set of candidate terms. These statistical measures aim at determining whether or not a given candidate term might be a true term and how related to the domain it might be. Statistical measures can be classified in two groups considering their final goal: measures aimed at determining the *unithood*, i.e., the degree of strength or stability of syntagmatic combinations and collocations to form a linguistic unit, and measures for the *termhood*, i.e., the degree to which a linguistic unit is related to the domain (Pazienza et al., 2005). *Unithood* measures, such as (Dunning, 1993; Fano, 1961; Salton et al., 1975) allow the recognition of complex linguistic units (called collocations) composed of words with a strong association, such as "day after" or "spacecraft mission". On the other hand, *termhood* measures determine the relatedness of the candidate terms with the domain.

For example, the Term Frequency-Inverse Document Frequency (TF-IDF) (Salton and Buckley, 1988) method combines the term appearances in a document with frequencies of the documents in which the term is found in a reference corpus to determine its *termhood*. On the one hand, the term frequency measures the relevance of the term. The more frequently a term appears in a document, the more relevant it is. On the other hand, the inverse document frequency measures the specificity of the term. The more documents the term appears in, the less specific the term is.

Other methods, such as Latent Semantic Analysis (LSA) (Deerwester et al., 1990), use more advanced statistical measures. LSA is a mathematical method for modeling the meaning of words and passages by analyzing representative text corpora. The Dirichlet Process Segmentation, which is a Bayesian method for non-parametric modeling, has also been recently applied for term extraction in (Koilada et al., 2012).

3.2.3 Hybrid Approaches

The syntactic patterns used for the identification of the terminology are language-dependent. Furthermore, they might recognize candidate terms which are not representative of the domain being described. Therefore, means to determine the *termhood* should also be used. Statistical approaches such as those described above are valid to address this objective. Hybrid approaches combine linguistic and statistical techniques for term extraction. They rely on syntactic patterns for detecting candidate terms and use statistical measures to determine their domain-relatedness and relevance.

Earl (1970) proposed one of the first hybrid systems, which firstly extracts noun-phrases as candidate terms and then ranks them according to the frequency of their noun elements. Daille (1994) proposed an approach in which the candidates terms are obtained using syntactic patterns and filtered using different statistical measures. Another similar approach is described in (Justeson and Katz, 1995), where expressions are used to extract the candidates, which in turn are ranked by frequency.

Enguehard and Pantera (1995) describe a more complex approach in which, in a first step, the terms are extracted according to their frequency. Then, in a second step, new terms are derived through linguistic heuristics applied to the terms retrieved in the first step.

A step further is to improve the linguistic analysis using semantic and contextual information. Maynard and Ananiadou (1999) derive semantic information from thesauri, linguistic hints and statistical evidence are mixed for ranking candidate terms. For example, the NC-value, a complex heuristic measure based on C-value, adds context factor information considering the semantic, syntactic and statistical properties of the context where the terms appear. This use of context information is also common in other approaches such as (Velardi et al., 2001), where a shallow syntactic parser is used to select candidate term patterns and, then, two measures –*Domain Relevance* and *Domain Consensus*– are used to rank the candidate terms, i.e., determine their *termhood*. The *Domain Relevance* measures the specificity of the candidate term with respect to the target domain, i.e., whether the term is exclusive of the domain or is broadly used in other knowledge areas, whilst the *Domain Consensus* refers to the homogeneous use of the candidate term in the domain. To compute both measures, collections of documents on each covered domain must be provided.

KEA (Frank et al., 1999; Medelyan and Witten, 2006) also relies on a hybrid process for the automatic extraction of keyphrases from documents. It first identifies a set of candidate terms (n-grams entailed by 1 to 3 words) and uses a Machine Learning algorithm to determine which candidates are good keyphrases. The Machine

Learning algorithm uses four features, the TF-IDF score, the number of words of the candidate keyphrase, the first occurrence of the candidate (computed as the percentage of the document preceding the first appearance of the term in the document), and the number of phrases the candidate set is related to.

GenEx (Turney, 2000) approaches keyphrase extraction from text as a supervised learning task. GenEx has two components, Genitor (Whitley, 1989), which relies on a genetic algorithm, and Extractor (Turney, 2000) that implements the keyphrase extraction algorithm. To fulfill its purpose, first the Genitor model has to be trained with procedural domain knowledge. Then, Extractor is used with those learnt parameters for keyphrase elicitation. The experimental results showed that the custom-designed algorithm designed for this task can generate better keyphrases than a general-purpose algorithm.

Hulth (2003) proposed a supervised Machine Learning approach in which linguistic knowledge (e.g., syntactic features) are used along with statistics (such as term frequency) for automatic keyword extraction. Hulth claimed that extracting noun phrases instead of n-grams increased the precision and including *part-of-speech* tags as features dramatically improved the keyword extraction performance.

HaCohen-Kerner et al. (2005) present an approach for eliciting keyphrases from scientific articles written in English. They combine different baseline methods similar to those used in summarisation –e.g., Kupiec et al. (1995)– and then applies common supervised Machine Learning methods in order to achieve the best combination of those baseline methods. In all methods, words and terms that have a grammatical role for the language are excluded from the key words list according to a ready-made stop list.

3.3 Extraction of Relationships

In the last years, several efforts have been conducted on Ontology Learning, including the elicitation of relationships. Maedche and Staab (2000) distinguished the identification of two main kinds of relationships while mining ontologies from text: taxonomic relationships (*isA*) and general or non-taxonomic relationships (e.g., *partOf*). Most efforts have addressed the extraction of taxonomic relationships but there are some approaches that have considered a larger set of relationships. Next, a review of different approaches is presented according to the kinds of relationships they deal with.

3.3.1 Extractors of Taxonomic Relationships

Ponzetto and Strube (2007) derived a taxonomy from the Wikipedia category system, the WikiTaxonomy. In order to build up such a taxonomy, they defined an algorithm for the elicitation of *isA* relationships from the Wikipedia categories. Syntactic patterns are used on those categories to infer the relationships. For instance, the head of the category allows identifying that a "British computer scientist *isA* Computer scientist". Algorithm uses additional patterns, such as Hearst patterns (Hearst, 1992), for the extraction of relationships. Finally, inference-based methods are applied to propagate the relationships identified in the previous step through the hierarchy, considering both multiple inheritance and transitivity. For example, as a *Leek* is known to be an *Edible Plant*, and an *Edible Plant* is a *Plant*, these methods would infer that "Leek *isA* Plant" taking advantage of the transitivity.

KOG (Wu and Weld, 2008), which stands for Kylin Ontology Generator, is an autonomous system that builds an ontology by combining the information in Wikipedia infoboxes with WordNet, using Machine Learning techniques. Infoboxes are tables containing meaningful information about the article in the form of attribute-value pairs in Wikipedia articles. Each infobox is considered a class and all its attribute-value pairs are represented by class slots. KOG uses Machine Learning, in particular a joint inference approach based on Markov logic, to infer *isA* relationships between pairs of classes. To this end, it uses several features such as: 1) similarity measures between the classes, 2) class-name string inclusion, 3) category tags, 4) whether or not the class-names appear in Hearst patterns in Google queries, and 5) WordNet mappings, which allows profiting from the defined hypernyms. For example, KOG would infer that "Earth *isA* Planet" from the "planets such as Earth" text fragment.

Flati et al. (2014) presented an approach for the automatic creation of an integrated taxonomy of Wikipedia articles/categories, i.e., a taxonomy of Wikipedia articles aligned to a taxonomy of categories. The main idea of integrating both article and category taxonomies is that this leads to a finer-grained taxonomy with higher coverage, the WibiTaxonomy. This enhanced taxonomy is built in three phases. First, an initial article taxonomy is built by parsing articles to extract the textual definitions which the articles include. The hierarchical taxonomy is generated by identifying hypernym relationships between the extracted definitions. Secondly, the system iterates over each extracted hypernym in the article taxonomy. Using the category links of each article, the category taxonomy is inferred. In each iteration of this algorithm, the links of the article taxonomy are used to discover category hypernyms, and these are used to discover more hypernyms. Finally, the category taxonomy is improved using structural heuristics, which will provide broader coverage to the taxonomy. For example, given an uncovered category c which does not have any connection to $c0$, being $c0$ the only direct super-category of c in Wikipedia, a link between them will be inferred.

MENTA (Melo and Weikum, 2014) is a multilingual taxonomy derived from Wikipedia. Unlike previous approaches, it was also built by analysing Wikipedia for languages other than just English. Therefore, it includes local information not covered by the English Wikipedia, such as local places, local people, local laws, etc. The information is organized into a coherent taxonomy using both Wikipedia and WordNet structures as references. To build the MENTA taxonomy, the following procedure was carried out. First, for each article, the parent categories are extracted. In addition, a small gloss, usually found in the first paragraph of the article, and the labels that are associated to the article are also gathered. The next step entails finding connections between all the gathered articles using several linking functions. For example, the cross lingual linker that will connect two articles where a link between articles of different Wikipedias exists. Another example is WordNet hypernyms, which defines connections between articles when the articles have a connection in WordNet. Finally, the last step involves aggregating the taxonomic information gathered and applying different filters to produce a clean taxonomy and make it even more consistent. For example, a filter that removes cycles of subclass relationships, given that all entities in the cycle are equivalent, is used to clean the taxonomy.

3.3.2 General Relationship Extractors

Nastase and Strube mined the Wikipedia category network to extract different types of relationships including *isA* and *partOf* relationships (Nastase and Strube, 2008). The novelty of their approach is that they focus specifically on using category names to extract the information and propagate this knowledge towards the articles connected to these categories. The process followed by the authors in order to extract relationships entails: 1) identifying the domain constituent of category names; for example, Chairmen of the county councils of Norway has three constituents: chairmen, county councils and Norway, being the dominating one chairmen, 2) extracting relationships from all the articles below the processed category, using syntactic patterns applied to the category name that will infer the relationship encoded in the category name. Figure 3.1 shows a complete example of the process. Albums by genre and Live blues albums categories are processed and the corresponding relationships are inferred connecting the articles Cookin' in Mobile and 12-String Blues with the corresponding extracted relationships (*isA* and *genre*).

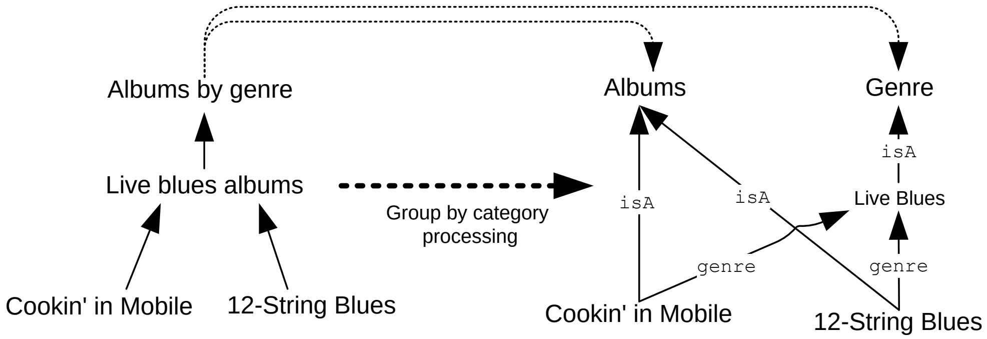


Figure 3.1 – Example of the Knowledge Extracted from WikiRelations

3.4 Additional Resources for Automatic Information Elicitation

In the last few years, many efforts have been conducted using additional resources of information such as those described below for automatic information elicitation.

3.4.1 Wikipedia

Wikipedia is a collaborative online encyclopedia containing over 35.9 million articles in 287 languages (as of August 2015)[1]. It has become one of the most popular reference works on the Internet. Wikipedia has a vast, constantly evolving tapestry of richly interlinked textual information (Milne and Witten, 2013). Therefore, it is a really powerful resource for NLP research or data mining, as it provides an ever-growing source of manually defined concepts and relations.

The article is the basic element of Wikipedia. An article, in Wikipedia, is identified by a unique name and contains information about a concept, an event or a relevant personality. Besides the content, the articles might also contain internal links to other articles and also external links. Articles are classified according to categories. A Wikipedia category provides a way to group related articles and add semantic knowledge to articles. A category has a unique name, and may have parent categories, child categories and articles belonging to the category.

Experts that want to use Wikipedia as a source of machine-readable knowledge have three options to choose from. They can rely on third-party secondary structures,

[1] `http://stats.wikimedia.org/EN/TablesArticlesTotal.htm`

such as Freebase (Bollacker et al., 2008), Yago2 (Hoffart et al., 2011) and DBPedia (Bizer et al., 2009). A second option would be to start from scratch and build their own algorithms for extracting the Wikipedia knowledge. Finally, a third option would be to develop and share the algorithms, rather than secondary resources.

The first approach is the easiest one, as the information obtained using the structures is already in a machine-readable format. Nevertheless, new innovations and mining techniques are introduced regularly, potentially rendering obsolete pre-built resources. Moreover, if these structures are not maintained periodically, such resources forego one of the greatest strengths of Wikipedia: its propensity to grow rapidly and adapt itself to the world's changes. The second option, working directly from the raw source, allows researchers to innovate and find new ways to mine knowledge from Wikipedia. The content of Wikipedia is released in the form of large XML dumps with cryptic markup that requires substantial efforts to build usable machine-readable data. The third option, which involves using a toolkit that helps to process the contents of Wikipedia to form machine-readable data, also provides an easy way to apply and share different techniques for gathering the knowledge contained in Wikipedia. This allows researchers to focus on the algorithms for knowledge extraction instead of dealing with the Wikipedia dumps.

3.4.2 WordNet

WordNet (Fellbaum, 1998) is a large lexical database initially built for English that has already been ported to various other languages, for instance, in the Multilingual WordNet (Bond and Foster, 2013). WordNet groups words (nouns, verbs, adjectives and adverbs) into cognitive synonyms called synsets. Each synset refers to a distinct concept that can be referred using different forms. Synsets are connected by semantic relationships such as hypernyms or hyponyms, being the synsets a noun.

WorNet has been used in the literature for a variety of different systems related to NLP including: Word Sense Disambiguation, Word Similarity measures (Agirre and Soroa, 2009), automatic text classification (Li et al., 2009), and even crossword puzzle generation (Aherne and Vogel, 2003).

3.4.3 Other Knowledge Bases Derived from Wikipedia

In the following lines, some resources for the relationship elicitation derived from Wikipedia are described. These resources have been developed to extract linked-data in order to use it in semantic applications.

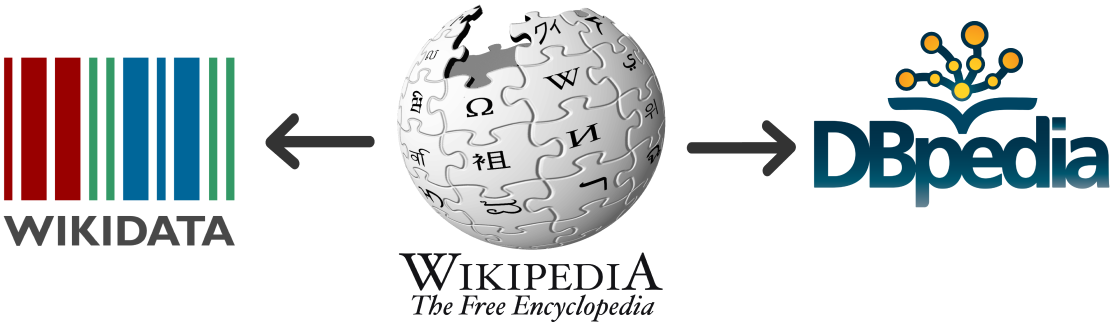


Figure 3.2 – Knowledge Bases Derived from Wikipedia

3.4.3.1 Yago

YAGO (Hoffart et al., 2013; Mahdisoltani et al., 2014; Suchanek et al., 2007) is a knowledge base derived from Wikipedia, WordNet, and GeoNames providing temporal and spatial information. Yago is an ontology composed of entities and facts that relate different entities between them (RDF tuples). An entity represents a concept or topic. Entities are extracted from Wikipedia articles, being each entity a Wikipedia article. Then facts, containing semantic information are extracted from each entity extracting information using the Wikipedia category system, Wikipedia Infoboxes information and WordNet Synsets. Each fact is annotated with a confidence value assigned by the different kinds of fact extractors.

Yago2 added some temporal and special abilities to Yago. On the one hand, in Yago2 the entities can be assigned to a time span denoting time existence. For example, "Elvis Presley has a birthdate of 1935-01-08 and a death date of 1977-08-16". Facts can have an assigned time point, for example, "population as of 2020" or a time span like the entities. On the other hand, Yago2 extracted spatial information about entities from GeoNames[2].

Yago3 gains from multilingual data from all the Wikipedias. For example, local places, such as small cities or towns, are usually defined in the Wikipedia edition that is attached to that place, then extracting entities from those Wikipedias improves the recall of the knowledge base. Yago3 does this by joining all the Wikipedia entities avoiding duplicates. That is, if we select the same Wikipedia article from different Wikipedias, only one entity will be generated combining the information extracted from all the Wikipedias.

[2] http://www.geonames.org/

3.4.3.2 DBPedia

DBpedia (Lehmann et al., 2014) is a knowledge base built on Wikipedia and it is a crowd-sourced community effort designed to extract structured information from Wikipedia and make this information available on the Web.

DBpedia extracts information from Wikipedia categories and infoboxes from the articles contained in each Wikipedia language edition. Each Wikipedia article gets an entity in DBpedia. The semantic knowledge for each entity is stored in the form of RDF tuples. These tuples are extracted from the Wikipedia Category system and for the infoboxes that are part of the Wikipedia articles. In order to have a common template for all the Wikipedia editions, a so-called DBpedia ontology has been manually created arranging the most commonly used infobox templates within the English edition of Wikipedia and then, mapping these template attributes to ontology properties.

DBpedia releases all this information as open source and tries to stay up-to-date with the Wikipedia using live snapshots containing the differences between different dates of each snapshot. DBpedia is not editable; in its current form it only contains information extracted from Wikipedia and its goal is to stay as close as possible to Wikipedia.

3.4.3.3 Wikidata

Wikidata (Vrandečić and Krötzsch, 2014) is a semantic knowledge database which is derived from Wikipedia. Wikidata obtains its data from all the Wikipedias, trying to reconcile the 287 language editions. For example, all the Wikipedia pages that represent *Rome*, the capital of Italy, are gathered in one item in Wikidata, the different language pages being linked to this item. To represent the semantic data (possible relations in our case) Wikidata employs property value pairs. For example: *Rome* might have a property *population* with value *2,777,979*. Every asserted property has its corresponding Wikidata page and is not linked to a Wikipedia article. Each property has a specific datatype that defines the value types, in the *Rome* example the datatype will be *number*.

Property value pairs are not enough to express the knowledge contained in Wikipedia. For example, Wikipedia says the "population of Rome is 2,651,040" (as of 2010). And this cannot be expressed easily with only property value pairs. Qualifiers subordinated to every property were introduced to state contextual information (such as the validity time for an assertion). They can also be used to encode ternary relations that elude the property-value model; for example, to say "Meryl Streep played Margaret Thatcher in the movie The Iron Lady", one could add to the item of the movie a property *cast member* with value *Meryl Streep* and an additional qualifier $role = MargaretThatcher$. These qualifiers in their current form are extracted from the data found in Wikipedia infoboxes and are validated using the information from the sources (references) of each article.

Finally, Wikidata also allows for two special types of statements. On the one hand, it can be specified that the value of a property is unknown. On the other hand, a property might not have value at all.

Wikidata is a relatively new project and not many systems are using it as a knowledge base. However, it is expected that systems such as those that build on DBpedia will start using it soon. Wikidata way to work with multilingual data shows its potential for relation extraction, not only for English but for all the 287 languages supported by Wikipedia. The relations can be extracted filtering information from properties and qualifiers.

3.4.4 Cyc

Cyc (Guha and Lenat, 1990) is a knowledge database built on human efforts. Cyc is composed of concepts such as *Bill Clinton*, *Spain*, and collections composed of various concepts or relations (e.g., Trees containing all trees). Cyc semantic knowledge is built upon predicates such “Madrid is the Capital city of Spain” or “Every tree is a plant”. The most important predicates are those describing hyponym and hypernym relationships. Facts about concepts can be obtained using CycL, a query language based on Lisp.

Cyc is divided into subgroups called Microtheories (MT), which are organized in tree-like structures. For example, The *Geometry* MT inherits from *Math* MT. Cyc requires each MT to be free from contradictions in their predicates.

Cyc, like other knowledge bases, can be used for question answering systems such as the one developed by Cleveland clinic[3]. To this end, Cyc provides truth functions that allow answering queries such as determining if two concepts are siblings.

3.5 Conclusions

This chapter has presented the current trends and techniques in Ontology Learning, including the techniques for term and relationship extraction and diverse sources of information that are currently being used to this end. These techniques and resources

[3] http://www.w3.org/2001/sw/sweo/public/UseCases/ClevelandClinic/

can be useful for the elicitation of Educational Ontologies and, therefore, provide the basis for the work presented in the following chapters of this thesis.

4 Identification of Multilingual Terminology

The elicitation of multilingual terminology is an essential step for the construction of *Multilingual Domain Modules.* This chapter presents *LiTeWi*, a tool for the elicitation of terms for Educational Ontologies from electronic textbooks that uses Wikipedia as an additional source of information. The chapter starts by describing *LiTeWi* (Section 4.1). Then, the evaluation process followed to validate the proposal is presented (Section 4.2). Next, a comparison with other approaches is depicted (Section 4.3). The extendability of *LiTeWi* to support new languages is described next (Section 4.4), and it finishes with some conclusions (Section 4.5).

4.1 LiTeWi: a Multilingual Term Extractor for Educational Ontologies

The term extraction techniques described in Section 3.2 are aimed at the identification of the most relevant terms in a document and have been broadly used for Ontology Learning. The work here presented aims at facilitating the development of Educational Ontologies for TSLSs, in particular the extraction of the topics to be mastered by the students. *LiTeWi* (Conde et al., 2015), a multilingual term extractor that uses Wikipedia and combines diverse term extraction methods has been developed.

Figure 4.1 illustrates the combined term extraction approach carried out by *LiTeWi*. It entails three main steps: the identification of the candidates using diverse techniques, the combination and the refinement of the results to obtain the set of terms, and, finally, the mapping of the terms to other languages in Wikipedia. Currently, the techniques used for term extraction are TF-IDF, CValue, KP-Miner, and Shallow Parsing. Some of those techniques require a tuning-up phase. Therefore, *LiTeWi* has been firstly tested using the *Principles of Object-Oriented Programming* textbook (Wong and Nguyen, 2010), which consists of 67 pages and over 30,000 words as input. In addition, an English dump of Wikipedia (4,400,000 English articles - as of February 2014) covering the terminology of a huge number of domains has been used as a generic corpus.

4.1.1 Candidate Extraction

In the approach here described, the extraction of the candidate terms entails running several term extraction techniques in parallel, aimed at obtaining the pursued terms. In a subsequent process, the unwanted terms will be filtered from the candidate list. The extraction of candidate terms entails running the algorithms with low thresholds where possible in order to identify as many terms as possible and to prevent discarding "real" terms. Next, the used term extraction techniques are described.

4.1.1.1 TF-IDF

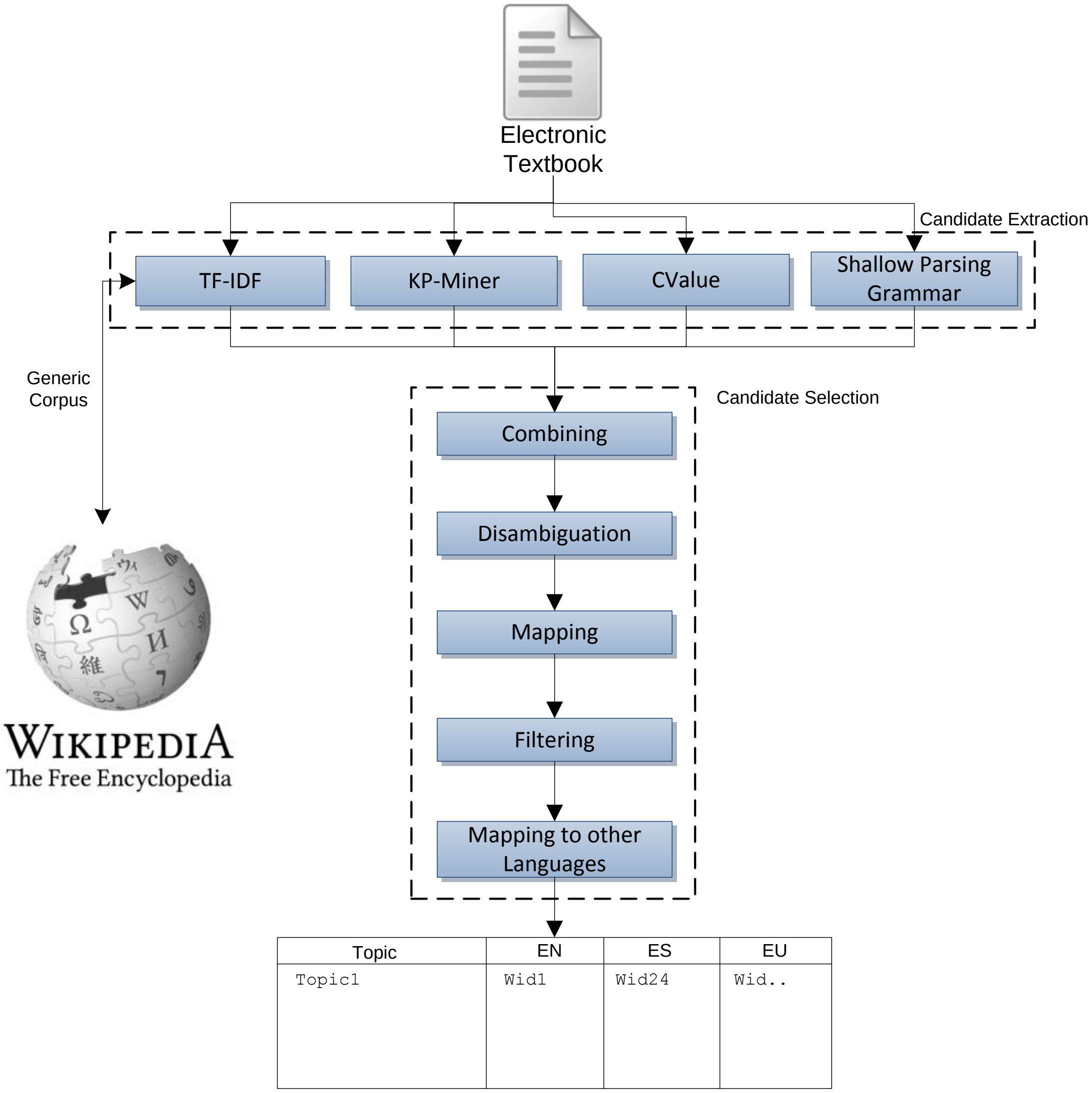
The TF-IDF (Salton and Buckley, 1988) technique for term extraction allows the identification of terms in a document. Besides the term frequency, this technique also considers the relevance of the terms in the document. TF-IDF requires a corpus to distinguish common terms from those which are relevant.

For processing Wikipedia, first a XML raw dump on Wikipedia must be downloaded and then, the content and the titles of the articles are extracted using the *Wikimedia Extractor*[1]. After extracting all the articles along with their content, the term frequencies for each article in Wikipedia must be calculated. This process is time-consuming, due to the huge amount of data to analyse. To accelerate the process, an *Apache Lucene*[2] index, a high performance search engine, has been used to process, store and elicit all the required information for processing the algorithm.

As can be observed in Figure 4.2, Wikipedia entails articles of different granularity or length. Given that short articles might refer to a very limited set of topics, they might considerably affect the performance of the TF-IDF method. Therefore, the TF-IDF extractor was first tested on the *Principles of Object Oriented Programming* textbook to determine whether or not small articles, the use of *stopwords* (words that are very common and, therefore, in term extraction, are usually filtered out prior to

[1] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor

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



Topic	EN	ES	EU
Topic1	Wid1	Wid24	Wid..

Figure 4.1 – Overview of LiTeWi

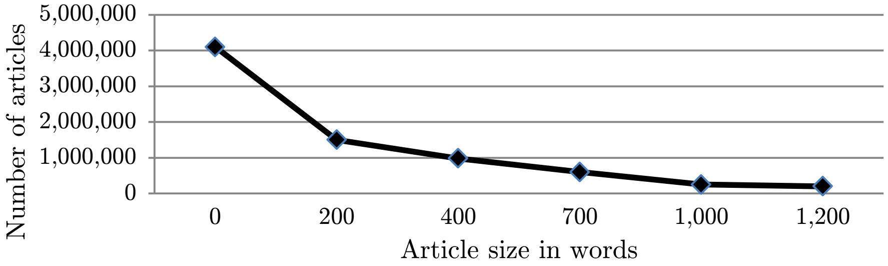


Figure 4.2 – Number of Wikipedia Articles vs. Article Size in Words

the process) or the stemming (reducing inflected words to their stem) might affect its performance. Concerning the size of the article, the best results were obtained when filtering articles with a length lower than 700 words.

The use of *stopwords* did not affect the performance, as those words tend to appear in almost every document and, therefore, have very low scores (see Table 4.1). For instance, words such as *"he"*, *"from"* or *"his"* usually have a high term frequency, but are used in most documents; therefore, they are not considered representative and obtain a low TF-IDF score avoiding the need to build and test an appropriate *stopword* list. However, the default stopword list for English used by *Apache Lucene*, which entails 33 words (see Appendix B.1), is applied in order to reduce the size of the Lucene index.

Table 4.1 – Top Terms for English in Wikipedia

Term	**Term Frequency**	**Document Frequency**
he	8,241,073	1,222,034
from	6,994,944	1,941,597
his	6,731,768	1,183,499
were	4,407,804	1,081,245
which	4,150,383	1,416,728
also	3,503,109	1,414,638
has	3,417,285	1,289,822

Stemming was discarded as it negatively affected the performance of the method. Given that some important word variants are converted to the stemmed word, they are lost. For example, *"Abstraction"*, which is a relevant topic in programming, was converted to *"Abstract"* using stemming. As *"Abstract"* is a common word in the Wikipedia corpus, it was discarded because of its score. In addition, a filter that removes the possessive genitive ("'s") is applied.

4.1.1.2 CValue

The CValue (Frantzi et al., 2000) is a domain-independent technique for extracting nested terms. It relies on statistical (frequency) and linguistic information, and takes into account the occurrence of terms candidates as a part of longer terms. In the work here described, a Java version of the algorithm was developed and employed. This implementation of the CValue relies on the *part-of-speech* provided by the Illinois Part of Speech Tagger[3] obtained in the preprocess of the analysed document.

CValue requires a linguistic filter to choose the terms to be weighted from the processed texts. Different linguistic filters lead to different results, affecting the *precision* and *recall* of the output list. Linguistic filters can be classified into two types (Mima and Ananiadou, 2000):

- *Close filters*, which are strict about the text fragments they permit. For example, Dagan and Church (1994) used a filter that only allows sequences of nouns (e.g., $Noun^+$).
- *Open filters*, which are more flexible and accept several kinds of strings. This kind of filters may have a negative effect on *precision* but will be positive in terms of *recall.* Justeson and Katz (1995) used open filters such as $Noun^+$, $(Adj|Noun)^+(Adj|Noun)^*$ and $(Noun|Prep)^?(Adj|Noun)^*)Noun$ for term extraction.

Table 4.2 – CValue Results Using Different Filters

Filter	Precision(%)	Recall(%)
$Noun^+$	10.48	25.17
$(Adj\|Noun)^+(Adj\|Noun)^*$	7.67	30.39
$(Noun\|Prep)^?(Adj\|Noun)^*Noun$	6.84	33.5

To determine which filter should be used, different filters were tested (see Table 4.2) on the *Principles of Object-Oriented Programming* textbook. Given that, at

[3] http://cogcomp.cs.illinois.edu/page/software_view/POS

this stage of the process, the goal is to maximize the *recall*, the $(Noun|Prep)^?(Adj|Noun)^*Noun$ open filter, which yields the best results, was chosen. Some examples of the term candidates identified by the CValue method are shown in Table 4.3.

Table 4.3 – Example of Extracted Terms with their CValue

Term	CValue
design pattern	28.41
list structure	23.5
variant behaviour	17
invariant behaviour	16
concrete subclass	15.75
gui component	15
concrete implementation	13

4.1.1.3 KP-Miner

KP-Miner (El-Beltagy and Rafea, 2009), which stands for keyphrase miner, is a system for the extraction of Arabic and English keyphrases from both text or html documents. Unlike other existing keyphrase extraction systems, KP-Miner does not require any prior training to fulfill its task. El-Beltagy and Rafea (2009) reported that KP-Miner produced comparable results for English for both KEA (Frank et al., 1999) and Extractor (Turney, 2000), two of the most broadly used keyphrase extraction systems. Examples of extracted terms using this system are illustrated in Table 4.4.

Table 4.4 – Example of Terms Extracted by KP-Miner

"object", "java", "drjava", "abstraction", "invariant","computer", "variant"

4.1.1.4 Shallow Parsing

The final goal of the approach presented throughout this paper is the extraction of terms for Educational Ontologies, learning topics, for which educational material can be found in the analyzed document. In a previous work, Conde et al. (2012) defined the *Didactic Resource Grammar*, a grammar that represents the most common syntactic structures used in DRs (e.g., *definitions, examples* or *theorems*). The *Didactic*

Resource Grammar, which is implemented using the *Constraint Grammar* formalism (Karlsson et al., 1995), was developed to enable the automatic extraction of LOs from electronic documents. An adapted version of the *Didactic Resource Grammar* is used by the Shallow Parsing method to identify fragments of the document that might contain DRs. The grammar consists of 250 rules: 145 rules that try to extract terms from *definitions*, 59 rules that will try to extract terms from *examples*, and 46 rules to extract terms from *principle statements*. The Shallow Parsing is carried out in two main steps: first, the text fragments containing potential DRs are filtered using the grammar. Then, noun phrases are extracted from those text fragments (see Algorithm 1) using the Illinois Chunker[4]. Those noun phrases entail the candidate terms.

ALGORITHM 1: Shallow Parsing Algorithm for Term Extraction

Input: Tokenized(POS-tags) Sentence List β, A grammar λ to apply
Output: A list of terms δ extracted applying the grammar to the sentences.

```
$termList$ = new List;
for $each\ tokenized\ sentence\ \alpha\ in\ sentence\ list\ \beta$ do
    $candidateSentence = applyGrammarToSentence(\alpha,\lambda)$;
    if $(candidateSentence.hasRuleMapped)$ then
        $nounPhrasesSentence = extractNP(candidateSentence)$;
        $term$=extractTermRule($nounPhrasesSentence$,
        $candidateSentence.ruleApplied$);
        $termList.addTerm(term)$;
    end
end
return $termList$;
```

Table 4.5 shows some examples of terms, highlighted in bold font, that were identified in a sentence selected by the grammar along with the rules that allowed their elicitation.

4.1.2 Candidate Selection

After running all the term extraction methods mentioned above, the results are combined as follows. First, all the results are combined in a huge term list. Then, each term is mapped to one or more Wikipedia articles. Next, the terms with more than one sense/meaning, i.e., more than one mapped article, are disambiguated.

[4] `http://cogcomp.cs.illinois.edu/page/software_view/Chunker`

Table 4.5 – Example of Constraint Grammar Rules

Pattern	**Example**
`Concept+(NOT Prep.)+(is|are)+[Det.]`	**Java** *is a* programming language. **Objects** are the primary units used to create abstract models.
`Concept+(refer|refers)+(to)`	**Abstraction** *refers to* object oriented programming.
`(This|That)+(is|are)+ (called)+Concept`	*That is called* the **Green House Effect**. This is called an **Array List**.
`(what)+(is|are)+[Det.]+Concept`	*what are those* **astronomical observatories**.
`(is|are) [adverb] + (called|known as|defined as) + Concept`	This phenomenon *is known as* **dynamic reclassification**. We use what *is called* the **assignment operation**.
`Concept+(is|are)+(used)+[Det.]+..`	A **list** is used to store objects. Abstract notion of a container structure. A **stack** is used to model systems that exhibit LIFO insert/removal behavior.

Finally, those terms not related to the desired domain are filtered. These steps are depicted in more detail in the following subsections.

4.1.2.1 Combining Term Candidates

After all the techniques for term extraction have finished, the returned results (all the terms obtained by the techniques described above) are combined in a large list of terms. Then, the elicited terms are normalized to ignore case and number differences.

After all the terms are combined, the *stopword* list shown in Appendix B.2 is applied to filter out unwanted terms. This *stopword* list was constructed combining the proposals of Salton (1971) and Fox (1990).

4.1.2.2 Mapping Terms to Wikipedia Articles

In this step, the terms obtained in the previous step are related to their corresponding Wikipedia articles. This entails searching in Wikipedia to determine whether or not each selected term can be related to one or more Wikipedia articles, each one representing a possible sense/meaning of the term.

In Wikipedia, each article, besides the title, has a set of labels that represent different variants for the title name of the article. To map the candidate terms with the Wikipedia articles, Wikiminer (Milne and Witten, 2013) is used. Wikiminer was developed to process the Wikipedia database dumps to form machine-readable data. Wikiminer is a platform where mining techniques take advantage of the Weka (Hall et al., 2009) Machine Learning workbench and the power of distributed computing using Hadoop (White, 2010).

Wikiminer processes Wikipedia dumps to build a database with information about articles, categories, links, labels, etc. As processing the Wikipedia dumps requires high computer resources, Hadoop can be used to take advantage of distributed computing. Wikiminer also provides a set of algorithms that allow data searches and comparisons to be performed. The functionality of Wikiminer is achievable through a set of web services. In Figure 4.3, the general architecture of Wikiminer is shown.

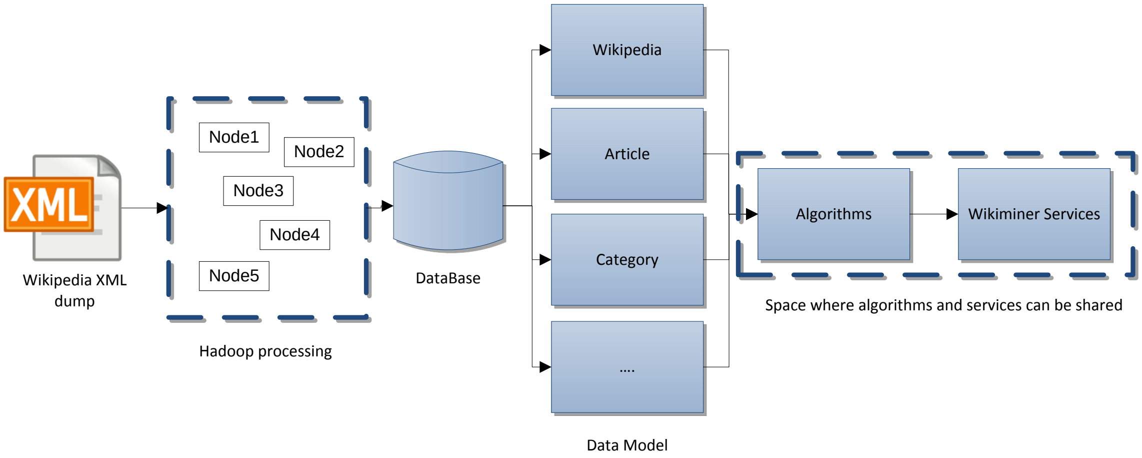


Figure 4.3 – General Architecture of Wikiminer

To fulfill the mapping task, Wikiminer uses both the article title and the set of labels. Some of this set of labels with their respective Wikipedia articles can be seen in Table 4.6.

Wikiminer requires some prior configuration to carry out its work. Some tests were conducted to determine the best method for mapping terms considering both *recall* and *precision*. Three different configurations have been tested. The first one just ignores case differences. The second one uses the Porter Stemmer (Porter, 1997). Besides ignoring case differences, it also removes possessive genitive cases. Finally, the third one ignores case differences, removes the possessive genitive, and uses the Pling Stemmer from Yago2s Java Tools[5] in order to remove plural cases. In the set-up experiments, the first achieved the best performance in terms of *precision* (29.34%). Surprisingly, its *recall* (55.26%) was quite satisfactory. The second one achieved 3.26% *recall* with 10.1% *precision*. The method using Pling Stemmer performed best, as it mapped 62.5% of the candidate terms with 25.35% *precision*. Therefore, this last alternative was selected.

Table 4.6 – Example of Labels for Different Wikipedia Articles

Wikipedia Title	**ID**	**Labels**
Java (island)	69336	Java, Javanese, Java Island, Island of Java, Jawa Dipa...
Java (programming language)	15881	Java Programming Language, Java, JAVA, java...
Earth	9228	Earth, earth, earth's, the Earth, global, planet Earth...
Solar System	26903	solar system, Sol system, Sol, star systems, the solar system...
Search for extraterrestrial intelligence	28153	SETI, S.E.T.I., Search for Extra-Terrestrial Intelligence...
List (computing)	208382	list, lists, Lists, list type, vector, sequence containers...

After this process, the terms are related to one or more Wikipedia articles. Those which are not related to any article are deleted. In the tests carried out, the size of the list is reduced by half. Furthermore, articles with the same sense/meaning or with the same list of senses/meanings are combined. To speed up the mapping step, a database that contains the normalized article names and labels has been built to use with Wikiminer. This database is used to compare the title names and labels with the candidate terms.

[5] `http://www.mpi-inf.mpg.de/yago-naga/javatools/`

4.1.2.3 Disambiguating the Terms

In the previous step, each term was related to one or more Wikipedia articles, each one representing a different sense/meaning. Therefore, a disambiguation process of the terms with more than one possible meaning is necessary. In the tests, a quarter of the terms needed disambiguation. Some examples of terms with their associated senses are shown in Table 4.7.

Table 4.7 – Example of Extracted Terms with their Possible Meanings

Term	Meanings
Java	island, programming language, software platform
GUI	graphical user interface, type of bowl-shaped Chinese vessel
Container	intermodal container (transport), abstract data type
light years	light-year, light years (Kylie Minogue album)
Einstein	Albert Einstein, einstein (crater)
keyboard	keyboard instrument, electronic keyboard, computer keyboard

A method that uses Milne and Witten Global disambiguation (Milne and Witten, 2008) approach is used to fulfil this task, to which end the Wikiminer Compare Service is used. This service provides a way for disambiguating term pairs using a classifier that takes as features:

- The data provided by Wikipedia. Wikipedia provides statistics about how an article label is associated to a sense/meaning. For example, 55% of *"Java"* labels refer to the programming language whereas 15% of them refer to the Indonesian island. These statistics yield three features for the classifier: the average, maximum and minimum prior probabilities of the two concepts.

- The semantic relatedness between the concepts. The relatedness score can be computed using the links of the articles as features. Milne and Witten (2013) claim that "*Wikipedia articles reference each other extensively, and at first glance the links between them appear to be promising semantic relations. Unfortunately, the article also contains links to many irrelevant concepts (e.g. terms not related to the domain of the analyzed book). Therefore, an individual link between two Wikipedia articles cannot be trusted*". There are different possibilities for computing the relatedness measure, for instance, using the article

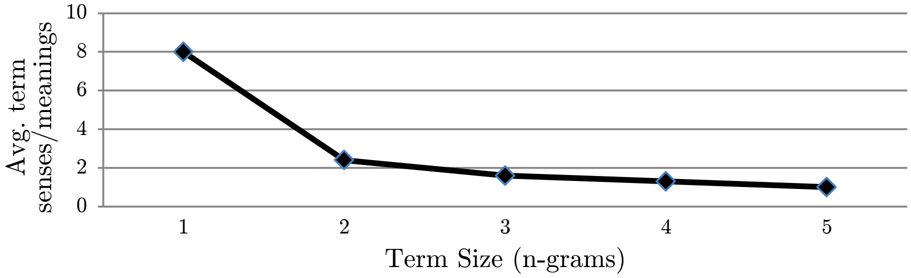


Figure 4.4 – Term Size (n-grams) vs. Average Number of Senses/Meanings

in-links (those links that refer to the article) and the article out-links (those links that are inside the article and refer to other articles). Both measures use different sets of links. The normalized distance measure is based on an approach that looks for documents that mention the terms of interest, and has been adapted to use the links made to articles. The vector similarity measure is based on an approach that looks for terms mentioned within two documents of interest, and has been adapted to use the links contained within articles. However, there is no reason why each measure should not be applied to the other link direction. Thus, each of the measures described above yields two features, one for in-links and the other for out-links. Finally, another measure taking into account the link counts for each article could be used. Different configurations have been tested. As pointed out by Milne and Witten (2013), the more features used, the higher the performance is. Therefore, the measure that combined the links-in, links-out and link-counts was selected for computing the relatedness score.

Each element of the candidate term list is disambiguated, following the approach summarized in Algorithm 2, to obtain its most plausible sense. The Wikiminer Comparing Service is used to fulfill such a task, to which end it requires a list of gold terms (terms with a unique meaning and that are relevant to the domain). But, how does one choose terms that are relevant to the domain and which have a unique meaning? Longer terms might be expected to be related to fewer articles. An analysis was conducted on the test results to determine whether or not the hypothesis was correct. As can be observed in Figure 4.4, the number of senses/meanings decreases as the term size in n-grams (number of words composing the term) increases.

Therefore, the more n-grams a term has, the more specific it is. Nevertheless, domain relevant terms are required. Hence, the monosemic terms with highest CValue score are chosen for the gold term list. This decision has been taken after making

ALGORITHM 2: Disambiguation Algorithm Given a Gold Term List

Input: Gold term list β, term λ to relate to the domain

Output: If the $term_\lambda$ is related to the domain returns the $term_\lambda$, if not returns *null*

```
for each term $\alpha$ in the gold term list $\beta$ do
    $sense, probability = wikiminerCompare(term_\alpha, term_\lambda)$;
    $term.addProbableSense(sense, probability)$;
end
List $probableSense_\delta = term_\alpha.getSenseList()$;
for each probableSense $\alpha$ in the probable sense list $\delta$ do
    $average$=calculateAverage($probableSense_\alpha$);
end
return $term_\alpha.maxAvgSense()$;
```

some tests with the CValue and observing that the top scored terms are almost always relevant in the domain.

Once the disambiguation finishes, an additional process is carried out to identify and combine terms that have been mapped to the same Wikipedia article.

4.1.2.4 Filtering Domain Related Terms

In this step, those terms which are not related to the domain are deleted. For this task, the gold term list built in the disambiguation step is used. This task attempts to relate each elicited term with the terms in the gold term list, to which end the Wikiminer Comparing Service has been employed. Again, this service has been configured to rely on the in-links, out-links and link-counts to determine the domain-relatedness.

The candidate term list is filtered following the process described in Figure 4.5. First, the Wikiminer Comparing Services computes each term domain-relatedness. Those topics whose score is below the threshold are dropped. Finally, those terms which are related with at least the minimum amount of gold terms are selected. Some experiments were conducted to determine the optimal thresholds and the number of “gold terms” that the candidates should be related to (at least one term, two terms or three terms).

As can be observed in Figure 4.6, the best results were obtained when requiring the candidate term to be related with at least one of the gold term list entries, with a relatedness score over 0.6. Therefore, this is the set-up that achieves the best compromise between recall and precision.

The algorithm for relating a term to the domain is presented in Algorithm 3.

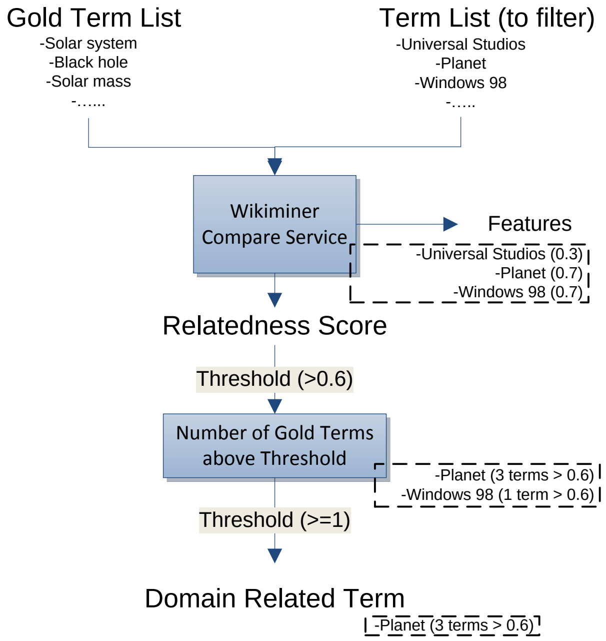


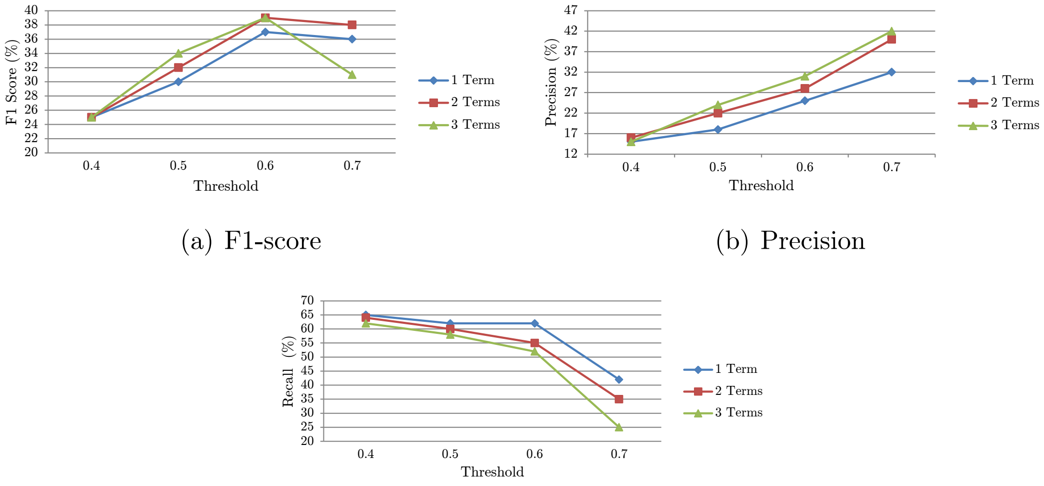
Figure 4.5 – Overview of the Filtering Algorithm

4.1.3 Mapping to Other Languages

In the last step, the final set of topics are mapped to their translations in other languages. To this end, *LiTeWi* profits from the connections that Wikipedia articles have to their equivalent articles in other languages. Given that in the previous steps each topic has already been mapped to its corresponding Wikipedia article, relating them to their translations is quite simple.

4.2 Evaluation

In the work here presented, the term extraction process has been evaluated using both *Gold-standard* and an expert validation. The evaluation was carried out on two books of different domains. The index of each analysed textbook has been used as *Gold-standard*. In addition, the elicited terms have been manually analysed by experts to determine whether or not the terms are related to the domain.



(a) F1-score

(b) Precision

(c) Recall

Figure 4.6 – Performance Regarding Threshold Values

The first book used for the evaluation is the *Introduction to Astronomy* (Morison, 2008) textbook. This book consists of 150 pages of plain text and over 110,000 words. The index is composed of 378 unique terms of which 114 are single word terms (1-grams), 189 terms are 2-grams, 57 terms are 3-grams, and 18 terms are 4-grams. 322 (out of 378) of the index terms were related to one or more Wikipedia articles. That is to say, 85.18% of the terms refer to at least one Wikipedia article, such a proportion being the best *recall* achievable.

The second book used for the evaluation is the *Introduction to Molecular Biology* (Raineri, 2010). This book consists of 139 pages of plain text with over 70,000 words. The index is composed of 274 unique terms of which 116 are single word terms, 119 of them 2-grams, 35 3-grams, 3 4-grams, and 1 5-gram. For this textbook, 220 out of 274 of the index terms were related to one or more Wikipedia articles. Hence, the best achievable *recall* is 81.30%.

In each book, *LiTeWi* was tested on a three-step process. The candidate extraction process was evaluated as detailed in the next section. In this experiment, each term extraction technique was tested on its own, measuring the recall according to the *Gold-standard*. The results of the candidate selection process were also evaluated using the same procedure. Besides, the remaining terms were also evaluated using the expert validation method. The validation allows recognizing terms that the authors might not have considered relevant when organizing the textbook, but could be

ALGORITHM 3: Algorithm to Calculate Relatedness Given a Gold Term List

```
Input: Gold term list $\beta$, term $\lambda$ to relate to the domain
Output: If the $term_\lambda$ is related to the domain returns the $term_\lambda$, if not returns $null$
$thresholdA$ = Relatedness score threshold;
$thresholdB$ = Number of terms above thresholdA;
$relatedToDomain = 0$;
for each term $\alpha$ in the gold term list $\beta$ do
    $relatedness = wikiminerRelate(term_\alpha, term_\lambda)$;
    if $(relatedness > thresholdA)$ then
        $relatedToDomain + +$;
    end
end
if $relatedToDomain > threholdB$ then
    return $term$;
else
    return $null$;
end
```

interesting for developing an Educational Ontology. The validation has been carried out by three experts. To determine the domain relatedness of candidate terms, only those terms that were considered valid by all the experts were selected. Finally, the mapping of the extracted terms to other languages was also measured.

As pointed out above, before carrying out the evaluation of *LiTeWi* on two books, it has been tuned-up on the *Principles of Object-Oriented Programming* textbook (Wong and Nguyen, 2010).

4.2.1 Results of the Candidate Extraction

The performance of the selected techniques is summarized in Table 4.8. The TF-IDF process identified 2,533 terms achieving 18.9% *recall* with 2.9% *precision* for *Introduction to Astronomy*, whilst it achieved 17.15% *recall* with 4.26% *precision* for *Introduction to Molecular Biology*.

The CValue process extracted 2,058 candidate terms, 6.9% of them contained in the index and covering 37.5% of the index for the first textbook, and 31.75% of the terms, with 2.48% *precision*, for the second textbook.

The KP-Miner identified 18.9% of the terms for *Introduction to Astronomy* textbook with 7.8% *precision*. However, it performs remarkably worse for *Introduction to Molecular Biology*, where it could only elicit 3.9% of the terms with a poor *precision*.

Finally, the Shallow Parsing Grammar identified terms in sentences that might be part of DRs such as *definitions* or *examples*. For the first textbook, it gathered 267 terms of the terms in the *Gold-standard*, which entails 13.42% *recall*. This method achieved 19.1% precision considering the *Gold-standard*. For the second textbook, it gathered 2.18% of the terms with 7.22% *precision*.

Table 4.8 – Results of the Candidate Extraction Methods over the Tested Textbooks

Measure	***TextBook***	**Precision(%)**	**Recall(%)**	**F1 Score(%)**
TF-IDF	*Astronomy*	2.9	18.9	5.02
	Mol. Biology	4.26	17.15	6.82
CValue	*Astronomy*	6.9	37.5	11.65
	Mol. Biology	2.48	31.75	4.6
KP-Miner	*Astronomy*	7.8	18.9	11.04
	Mol. Biology	1.82	3.9	3.9
Shallow Parsing	*Astronomy*	19.1	13.42	15.76
	Mol. Biology	7.22	2.18	3.34

Some researchers have reported remarkable performances of the TF-IDF method, which achieved similar scores to those obtained by the CValue method on certain domains (Zhang et al., 2008). They pointed out that the performance of the algorithm might be influenced by the importance of the single word terms in the domain. In the analyzed documents, only 30% – 42% of the index topics were single word terms, which explains the poor performance of the TF-IDF in this experiment.

Being most of the topics multi-word terms, the multi-word term extraction methods might be expected to perform better in terms of *recall* than the TF-IDF. The results, which confirmed that intuition, are consistent with those obtained in (Frantzi et al., 2000; Koilada et al., 2012; Zhang et al., 2008). As can be observed, the CValue performed much better and showed the advantage of combining *termhood* and *unihood* for term extraction methods.

4.2.2 Results of the Candidate Selection

In this section, the evaluation of each step in the candidate selection process is presented.

4.2.2.1 Combining Term Candidates

Once the candidate extraction has finished, the results obtained with every technique are combined and duplicates removed. After this step, the candidate term sets entailed 12,279 candidates for *Introduction to Astronomy* and 17,201 for *Introduction to Molecular Biology*. As expected, combining the results of the algorithms increased the *recall* remarkably. However, the *precision* is further reduced as domain-unrelated terms, or even wrong terms, affect the *precision*. The drop in the *precision* was an anticipated effect, but the following steps will improve its score.

4.2.2.2 Mapping Terms to Wikipedia articles

As mentioned above, the candidate terms are related to the Wikipedia articles to determine their domain-relatedness and to later filter unrelated terms. Mapping the terms to Wikipedia articles reduced the term list from 17,201 terms to 6,574 items in the Astronomy textbook. Furthermore, 1,831 terms related to one Wikipedia article (meaning only one sense/meaning) were found in the Astronomy textbook. In the Biology textbook, the candidate term list shrank from 12,279 to 2,688 terms, being 880 of them related to only one Wikipedia article.

4.2.2.3 Term Disambiguation

In the next step, those terms related to more than one Wikipedia article were disambiguated. Moreover, those candidate terms that were mapped to the same Wikipedia article, i.e., they refer to the same meaning, were combined into one term. For the Astronomy textbook, the candidate term list was reduced to 3,972 terms, 295 of them included in the *Gold-standard*. However, 1,803 were considered domain-related by the experts. On the other hand, the term list of the Biology textbook was composed of 1,194 terms in total, being 174 from the index and 455 related to the domain of the textbook, as the experts had stated. Table 4.9 presents the *precision*, *recall* and *F1-Score* (the harmonic mean) of this step for both textbooks, considering the gold-standard and the expert validation. Given that the *recall* can not be measured using the validation approach, the corresponding cells contain the *non-applyable* (N.A.) value.

4.2.2.4 Filtering Domain Related Terms

In the final step, those terms that were not related to the domain were removed. A remarkably improved precision can be observed in both the index and the domain related terms while barely affecting the *recall* (see Table 4.10).

Table 4.9 – Results after Disambiguation

	Gold-standard			Exp. validation
TextBook	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	**Correctness (%)**
Astronomy	7.51	77.83	13.69	45.39
Mol. Biology	12.73	63.50	21.20	38.10

After this step, the resulting term list for the Astronomy textbook was composed of 1,545 terms, 275 of them included in the gold-standard and 1,217 of them related to the domain. On the other hand, the term list of the Biology textbook was composed of 635 terms, being 165 from the index and 455 related to the domain of the textbook. Table 4.10 summarizes the statistics of this process.

Table 4.10 – Results after Domain Termhood Processing

	Gold-standard			Exp. validation
TextBook	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	**Correctness (%)**
Astronomy	17.96	72.55	28.79	78.77
Mol. Biology	27.09	57.29	21.37	71.65

4.2.3 Results of the Mapping to Other Languages

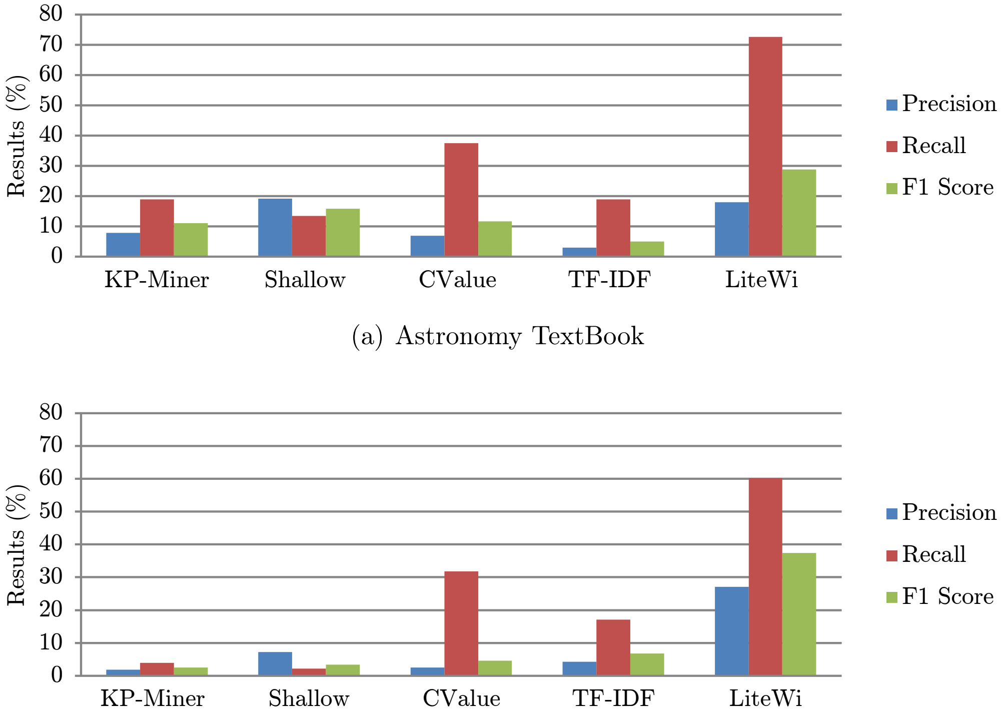
As the candidate terms have been mapped to Wikipedia articles, their translations to other languages can be elicited to build a Multilingual LDO. Table 4.11 shows the outcome of this step. For the Astronomy textbook, *LiTeWi* extracted 1,545 terms. 1,236 of them (80%) have a Spanish translation, 1,297 (84%) have a French translation, and 602 (39%) have a Basque translation. In the Biology textbook, *LiTeWi* elicited 635 terms. 476 of them (75%) have a Spanish translation, 469 (74%) have a French translation, and 203 (32%) have a Basque translation.

4.2.4 Results of the Overall Process

The results obtained by each technique along with the performance of *LiTeWi* are presented and compared in Figure 4.7.

English	Spanish	French	Basque
1,545	1,236	1,297	602
635	476	469	203

Table 4.11 – Number of Topics with Direct Translations to Other Languages



(a) Astronomy TextBook

(b) Biology TextBook

Figure 4.7 – Hybrid Approach vs. Other Algorithms

Comparing *LiTeWi* with each term extraction technique it uses, *LiTeWi* outperforms the best chosen technique (CValue) by over 30%, showing that this approach improves the results considerably.

As Hartmann et al. (2012) claim, there is a tendency to prefer hybrid term extraction methods that use the *termhood* and *unithood* measures as the CValue because of their superior performance. Nevertheless, there is no consensus on which the optimal method is. Some methods perform better on domains or type of corpus, whilst others are more successful on certain kinds of terms. *LiTeWi* provides an appropriate method which is valid for all these cases. It takes advantage of the chosen techniques to get as high a recall as possible, and then, using Wikipedia, it tries to improve the precision of the results by filtering unwanted terms. Some techniques perform better in certain domains than in others (see Figure 4.7). Given that *LiTeWi* combines several techniques, it collects all their results and has a more stable performance.

4.3 Comparison with Other Approaches

In this section, the approach presented here is compared with two statistical methods aimed at extracting multi-word terms and Wikifier, a state-of-the-art Entity Linking Tool. Entity Linking refers to the task of determining the reference of entities mentioned in a text within a knowledge base. The statistical approaches tested on this comparison are Point Wise Mutual Information (PMI) and Chi-Square (X^2). PMI evaluates the strength of the association between the words in a multi-word term candidate. On the other hand, the X^2 measures the significance of the association between the words in a multi-word term candidate. Both methods originally aim to extract bigrams but are adapted to longer terms in (Silva and Lopes, 1999).

In order to run PMI and Chi-Square, the NLTK Python Toolkit (Bird et al., 2009) has been used. The procedure for setting up the algorithms has been the same as that followed for *LiTeWi*. First, some empirical tests have been carried out to tune up the method using the *Principles of Object-Oriented Programming* textbook and, then, the evaluation with the two textbooks mentioned above has been performed.

As the algorithms are purely statistical, they return a lot of *"noisy"* terms, terms that do not make any sense. Then, the *stopword* list that can be found in Appendix B.2 (307 words) was applied to remove those terms. Besides, a minimum term frequency is required for multi-word terms to be selected; those terms with a frequency below 3 were filtered out.

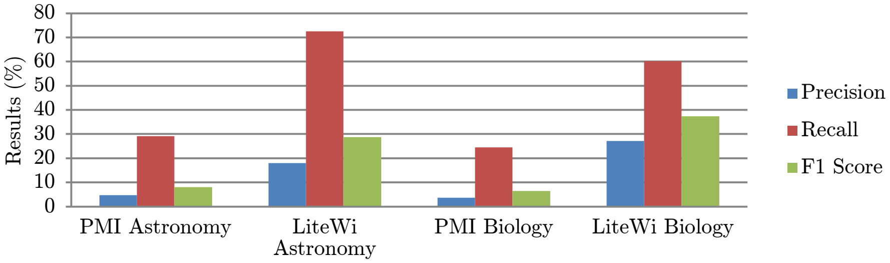


Figure 4.8 – LiTeWi vs. PMI

4.3.1 Pointwise Mutual Information

The Pointwise Mutual Information (PMI) (Fano, 1961), evaluates the strength of the association between the words in a multi-word term candidate. It takes into account the probability of observing n variables together (the joint probability) with the probabilities of observing those n variables independently (chance).

The results presented in Table 4.12 were obtained applying the PMI method. From the Astronomy textbook, a list of 2,340 terms was elicited, where 110 were part of the *Gold-standard* and 307 of them are related to the domain. In the case of the Biology textbook, 1,587 terms were extracted, 59 of them being part of the index and 193 of them related to the domain.

Table 4.12 – PMI Results

		Gold-standard		**Exp. validation**
TextBook	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	**Correctness (%)**
Astronomy	4.7	29.1	8.09	13.11
Mol. Biology	3.71	24.53	6.44	14.99

The comparison to *LiTeWi* can be seen graphically in Figure 4.8, where remarkable performance differences, with a 200% increase in *recall* and more than a 200% increase in *precision*, can be observed between the PMI technique and *LiTeWi*.

4.3.2 Chi-Square

Chi-Square (X^2) (Helmert, 1876; Plackett, 1983), measures the significance of the association between the words in a multi-word candidate. It allows the identification of sequences of words that occur together more than they might by chance, and, hence, can be considered as terms.

Processing the X^2 technique resulted in a term list composed of 2,011 terms, where 304 terms were related to the domain and 94 form part of the index for the Astronomy textbook. The term list for the Biology textbook is composed of 1,680 terms, of which 50 were included in the index and 193 were related to the domain. These results are described in Table 4.13.

Table 4.13 – X^2 Results

	Gold-standard			Exp. validation
TextBook	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	**Correctness (%)**
Astronomy	4.7	24.86	7.86	15.11
Mol. Biology	2.97	18.24	5.10	14.99

The comparison with the proposed hybrid approach can be seen graphically in Figure 4.9. Again, *LiTeWi* outperforms X^2 in terms of *recall* and *precision* by more than 200%.

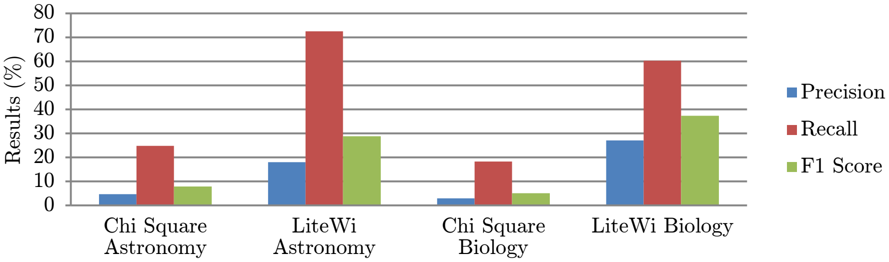


Figure 4.9 – LiTeWi vs. X^2

4.3.3 Wikifier

The Wikifier (Cheng and Roth, 2013; Ratinov et al., 2011) entity linking tool was developed to identify important entities and concepts in text, disambiguate them, and link them to Wikipedia. Wikifier follows these steps:

1. Identify which expressions must be linked to Wikipedia.

2. Disambiguate the ambiguous expressions and entities.

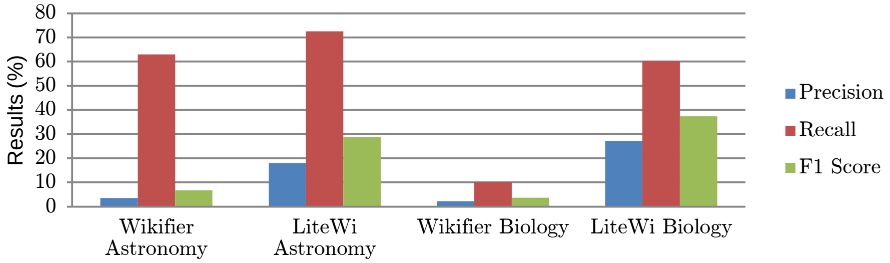


Figure 4.10 – LiTeWi vs. Wikifier

Both steps are similar to those made by *LiTeWi* for the same purposes, except that Wikifier requires a training corpus for both steps. Wikifier achieved 62.96% *recall* on Astronomy, whilst this score dramatically dropped to 10.21% on Molecular Biology (see Table 4.14). In both textbooks, the *precision* was very low. Regarding the domain-relatedness, it achieved 18.55% *precision* whereas it performed much better on Molecular Biology (49.27%).

Table 4.14 – Wikifier Results

	Gold-standard			Exp. Validation
TextBook	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	**Correctness (%)**
Astronomy	3.55	62.96	6.72	18.55
Mol. Biology	2.24	10.21	3.67	49.27

As can be observed in Figure 4.10, Wikifier obtained slightly worse results to those achieved by *LiTeWi* on Astronomy. However, *LiTeWi* performed remarkably better on Molecular Biology. The poor results in Biology may be related to the nature of how Wikifier has been trained to detect which expressions should be linked to Wikipedia.

4.4 Extendability of LiTeWi to New Languages

LiTeWi has been designed to deal with different languages. The prototype described and evaluated throughout this chapter works on documents written in the English language. However, including the support to deal with documents written in a new language in *LiTeWi* entails minor changes.

Obviously, and as reported in Section 2.3, a NLP parser for the new supported language must be integrated in *LiDom Builder* in order to facilitate the document preprocess before the subsequent knowledge extraction steps. In addition, some resources (e.g., patterns for the term elicitation) must be defined for the new language. Some of the extractors rely on language-dependent syntactic patterns that must be adapted (e.g., CValue and Shallow Parding). Besides, the candidate selection requires *LiTeWi* to map the topics to Wikipedia articles to determine which topics must be filtered. To this end, Wikiminer is used. Therefore, Wikiminer has also to be trained for the new language in order to compute relatedness measures.

In particular, *LiTeWi* has already been extended to support the extraction of multilingual topics from documents written in Spanish. To this end, a NLP parser that support Spanish, FreeLing (Padró and Stanilovsky, 2012), has been integrated in the prepocess. In addition, the grammars for the Shallow parsing and the CValue have also been defined and Wikiminer has been trained on the Spanish dump of the Wikipedia to facilitate the computation of the relatedness measures.

4.5 Summary

This chapter has presented *LiTeWi*, the module responsible for the term extraction in *LiDom Builder*. *LiTeWi* carries out a three-step process for the extraction of multilingual terminology from an electronic textbook, to which end it uses Wikipedia as an additional resource. *LiTeWi* firstly identifies a set of candidate terms using a bunch of techniques and tools, namely TF-IDF, CValue, KP-Miner and Shallow Parsing. Next, the candidate terms are filtered according to their domain relatedness, to which end *LiTeWi* relies on Wikipedia. Finally, it takes advantage of the multilingual nature of Wikipedia to link the final set of topics to their translations or equivalents in the supported languages.

To assess its performance, *LiTeWi* has been tested on two books of different domains: *Introduction to Astronomy* (Morison, 2008) and *Introduction to Molecular Biology* (Raineri, 2010) after tuning it up with *Principles of Object-Oriented Programming* textbook (Wong and Nguyen, 2010). The promising results of this evaluation have been also presented throughout this chapter. Finally, the extendability of *LiTeWi* to deal with the extraction of multilingual topics from documents written in a new language has also been highlighted.

The next chapter presents the elicitation of pedagogical relationships for Educational Ontologies.

5 Identification of Pedagogical Relationships

In this section two different tasks are presented. First, the automatic identification of structural relationships from document outlines written in English is addressed to confirm the language independence of the approach proposed in *DOM-Sortze* (Section 5.1). In addition, the benefit of incorporating Wikipedia as a knowledge source in the elicitation process is also considered. Then, *LiReWi*, a module for the elicitation of pedagogical relationships from the document body is presented (Section 5.2). Again, additional resources such as Wikipedia and Wordnet are employed. Next, the evaluation of the proposal (Section 5.3), and the comparison with other approaches are presented (Section 5.4). The chapter concludes with some final remarks (Section 5.5).

5.1 Testing Language Independence in the Elicitation of Structural Relationships

As pointed out above, in the approach presented throughout this thesis, the *Domain Module* of a TSLS is described by means of an Educational Ontology, the LDO, and LOs. In *DOM-Sortze*, the LDO contains the main domain topics and the pedagogical relationships among them. Pedagogical relationships can be structural –*isA* and *partOf*– or sequential –*next* and *prerequisite*.

DOM-Sortze was developed under the assumption that pedagogical information underlies the document structure. Using patterns, eliciting pedagogical relationships from the document outline and the document body is possible. Following this assumption, the analysis of the document outline was extended to deal with documents written in languages other than Basque. This section presents the process carried out to identify structural relationships from document outlines in English (Conde et al., 2014).

5.1.1 Identification of Structural Relationships from Document Outlines

The outline analysis process consists of two phases (see Figure 5.1):

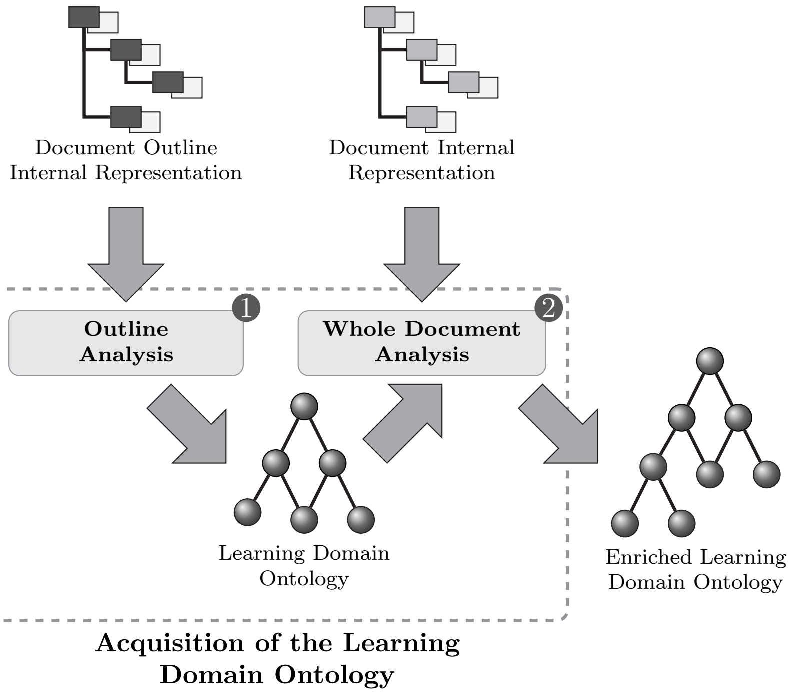


Figure 5.1 – Acquisition of the Learning Domain Ontology

- **Basic analysis**. In this task the main topics of the domain and the relationships among these topics are mined from the outline. In this approach, each index item is considered as a domain topic. Besides, the structure of the document outline is used as a means to gather pedagogical relationships. A subitem of a general topic is used to explain part of it or a particular case of it. Therefore, structural relationships are defined between every outline item and all subitems.

- **Heuristic analysis**. The results of the basic analysis are refined based on a set of heuristics that categorize the relationships identified in the previous step and mine new ones. The heuristics entail the condition to be matched, and the postcondition, i.e., the relationships that are recognized. The heuristic analysis relies on the empirically gathered confidence on the heuristic, i.e., the percentage of times the heuristic fires correctly.

The identification of structural relationships is carried out to categorize the relationship between an item and its subitems. In previous experiments, it had been noticed that the *isA* relationships could be inferred in different cases (see Table 5.1). On the one hand, homogeneous subitems, such as those shown in Table 5.1, allow the identification of such a structural relationships. Both subtitems share a common head (**clustering**) which is enhanced with some modifier following a *Genus et differentiam* pattern. This pattern is a means to present definitions which define a species (that is, a type –not necessarily a biological category) as a subtype of a *genus* satisfying certain conditions (the *differentiam*). In the example, both subitems show numerical classification methods. A set of heuristics (*group heuristics*) allow the identification of *isA* relationships from those kinds of structures. On the other hand, other fragments containing *isA* relationships are more heterogeneous. In the example, three kinds or security methods are presented. The first one is an *acronym* whereas the second one is a *proper name. Individual heuristics* are aimed at the identification of structural relationships in these situations.

Homogeneous subitems	Heterogeneous subitems
5.2 Numeric classification	6. Transport and network-level security methods
5.2.1 Exclusive **clustering**	6.1 **SSL**
5.2.2 Hierarchical **clustering**	6.2 **IPSec**
	6.3 Virtual private networks

Table 5.1 – Examples of Outline Fragments from which IsA Relationships Can Be Inferred

The structural relationships are identified in a heuristic-driven process that applies the algorithm shown in Algorithm 4. For each outline item, a group heuristic that matches is looked for. Group heuristics identify *isA* relationships from homogeneous subitems (see Table 5.1) or if the outline item entails certain keywords. If such a heuristic fires, then a *isA* relationship is defined between the outline item and each of its subitems. Otherwise, the individual heuristic that triggers is searched for on every subitem. Different heuristics can be fired together in the same group of subitems, so, the most confident one is returned; the *default* heuristic (*partOf*) is returned when no other heuristic condition is met (Larrañaga et al., 2004). Then, the list of applied heuristics is processed to get the confidence on an underlying *isA* relationship using (Equation (5.1)),

$$conf_{isA} = \frac{\sum\limits_{h \in H_i} f(h) \cdot c(h) - \sum\limits_{h \in H_p} f(h) \cdot c(h)}{n} \tag{5.1}$$

where h represents a heuristic, $f(h)$ is the number of times the heuristic h is triggered, $c(h)$ is the confidence on heuristic h, H_i the set of heuristics that identify *isA* relationships and H_p the set of heuristics that reinforce the hypothesis that the relationship is a *partOf* relationship, and n represents the number of subitems. If the $conf_{isA}$ value goes beyond a *threshold*, then the structural relationships are refined as *isA*, otherwise, the relationships are labeled as *partOf*. As mentioned above, every relationship is labeled with information about the heuristic that has been used to infer it.

ALGORITHM 4: Algorithm for Identifying Structural Relationships for an Outline Item

```
$gHeur \leftarrow findGroupHeuristic(outlineItem)$
**if** $gHeur \neq nil$ **then**
  $applyGroupHeuristic(outlineItem, gHeur)$
**else**
  $hList \leftarrow new\ List()$
  $subItems \leftarrow getSubItems(outline)$
  **for all** $subit\ in\ subItems$ **do**
    $iHeur \leftarrow findHeur(outlineItem, subit)$
    $add(hList, iHeur)$
  **end for**
  **if** $conf_{isA}(hList) > threshold$ **then**
    $addIsARel(outline, subItems, hRel)$
  **else**
    $addPartOfRel(outline, subItems, hRel)$
  **end if**
**end if**
```

To support the acquisition of structural relationships from document outlines written in English, equivalent heuristics to those described in (Larrañaga et al.,

2004) have been defined. Those heuristics rely on syntactic patterns and do not use any domain-specific knowledge, i.e., they are domain-independent. Some of those heuristics rely on NLP services, for instance, those to identify entity names (see Table 5.1). Therefore, the NLP services have to provide the same functionality for English, to which end they have been enhanced to use the Illinois Named Entity Tagger (Ratinov and Roth, 2009) for NLP tasks. This tool has been mainly used for entity recognition. Table 5.2 shows an example of a heuristic, which relies on the use of the possessive genitive to identify *partOf* relationships, that has been adapted to English.

Table 5.2 – Example of a Heuristic Adapted to English

Basque	English
4.5 **Inplementazioa**	4.5 **Implementation**
4.5.1 Aplikazioa**ren inplementazioa**	4.5.1 **Implementation of** the aplication
4.5.2 Agente**en inplementazioa**	4.5.2 **Implementation of** the agents

5.1.2 Experiment

To validate the proposal, 57 outlines of different courses offered in English at the University of the Basque Country (UPV/EHU) have been processed. These outlines are classified into three main areas: (1) Social sciences and humanities, (2) Engineering and architecture, and (3) Sciences. 27 courses were related to the first main area, 18 to the second, and 12 to the last one.

The evaluation of the proposal here described was conducted following a *Gold-standard* approach. Some members of the Galan research group, in collaboration with the lecturers of the courses whose outlines were used on the evaluation, defined the LDOs that were used as optimal output. These LDOs were restricted to the topics referred on the outlines and the structural relationships between those topics (1197 *partOf*, 483 *isA*). Then, every outline was processed and the automatically gathered ontologies were compared to the *Gold-standard*. The process was evaluated in terms of *recall*, i.e., the percentage of identified relationships, and *precision*, i.e., the percentage of correctly classified relationships.

	Precision (%)	**Recall (%)**	**F1 Score (%)**
isA	78.95	21.20	33.42
partOf	84.12	98.66	90.81
Total	84.15	83.850	84.00

Table 5.3 – Results of the Acquisition of the Structural Relationships from Outlines

Table 5.3 shows the results of experiment. The overall *precision* and *recall* are satisfactory (83.85%). Furthermore, the scores achieved for the *partOf* relationships were even higher; 84.12% *precision* and 98.66% *recall*. However, the *recall* for *isA* relationships dramatically dropped to 21.20%, although the *precision* was still satisfactory (78.95%). A deep analysis of the results was conducted to determine why the results were much worse than expected, even for a domain-independent approach. The lack of knowledge on certain domains significantly affected the performance of the process. For instance, it was observed that many of the topics involved in the missing *isA* relationships contained *proper names* (e.g., names of illnesses on a clinical nursing course); however, the entity name recognizer used in the experiment was unable to identify them. A training process would be necessary to fulfil such a purpose. Given that the process aims to be domain-independent, this was not an option.

To improve the results, a new step was included in the structural relationship elicitation process using Wikipedia as an additional resource. This improvement is described in next section.

5.1.3 Enhancing the Process with Wikipedia

Wikipedia is an appropriate resource for NLP given that it is: domain independent (it has a large coverage), up-to-date, and multilingual (Ponzetto and Strube, 2007). In their work, Ponzetto and Strube (Ponzetto and Strube, 2007) derived a large scale taxonomy containing *isA* relationships from Wikipedia. In the proposal here presented, this taxonomy has been used to discover new *isA* relationships. This process is aimed at refining the LDO discovering missing *isA* relationships, which describe specializations of the existing LDO topics. In most cases, these kinds of relationship appear in lower-levels (involving leaf nodes) of the LDO. To improve the LDO gathered by the heuristic outline analysis, an additional process is carried out:

1. Identify groups of sibling nodes (topics) of the LDO extracted from the outline analysis.

2. The groups of leaf nodes in which the *partOf* relationship has been identified are selected for refinement in the subsequent steps.

3. In the next step, the system tries to map every node with its corresponding Wikipedia articles. To accomplish this task, nodes are first normalized (removing plural marks, apostrophes and avoiding case differences). Then, every node is linked to those articles which are labeled with the normalized text of the node.

4. As some nodes might be mapped to more than one article, a disambiguation process is applied so that every node is mapped to a unique article. Wikiminer (Milne and Witten, 2013) is used to fulfill such a goal.

5. Process every group, using Ponzetto and Strube's taxonomy (Ponzetto and Strube, 2007), to look for a common ancestor. The system infers *isA* relationships in those groups that share a common ancestor, as long as it is not a general topic, i.e., it does not appear at top-levels in the taxonomy.

	Precision (%)	**Recall (%)**	**F1 Score (%)**
isA	77.30	50.33	60.97
partOf	89.19	96.49	92.70
Total	87.30	87.30	87.30

Table 5.4 – Results of the Acquisition of the Structural Relationships from Outlines Using Wikipedia

The results of the enhanced process have also been tested using the *Gold-standard*. Table 5.4 shows the results using the Wikipedia-based refinement. As can be observed, the overall performance has improved (87.70% *precision* and *recall*). Regarding *partOf* relationships, the *recall* has slightly decreased (96.49% vs. 98.66%). However, the *precision* for such a kind of relationships has also slightly increased from 84.12% to 89.12%. As regards *isA* relationships, the *recall* has dramatically increased from 21.20% to 50.53% whereas the *precision* was hardly affected (77.30% vs. 78.95%). Therefore, it can be concluded that the use of Wikipedia is beneficial for the elicitation of structural relationships from document outlines.

In addition to structural relationships between domain topics, a LDO contains more types of relationships, such as ordering relationships. Moreover, besides document outlines, document bodies are also valid inputs for eliciting relationships between topics. Next, *LiReWi*, a relationship extractor for Educational Ontologies from

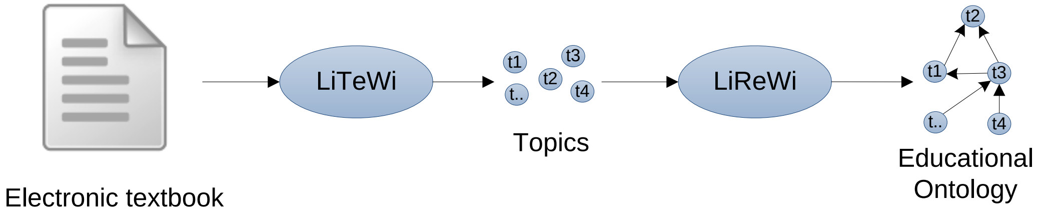


Figure 5.2 – Overview of the General Process

document bodies is presented. The main characteristics, along with the experiment conducted to validate the proposal, are depicted.

5.2 LiReWi: a Relationship Extractor for Educational Ontologies

Although the relationship extraction approaches described in Section 3.3 mainly address the identification of taxonomic relationships between the topics of an ontology, *LiReWi* (Conde et al., submitted) combines some of those approaches and sources of information to elicit a more complete set of pedagogical relationships (*isA*, *partOf*, *pedagogicallyClose* and *prerequisite*) from the document body. Again, *LiReWi* is intended to be usable on documents of any domain. Thus, any domain-dependent technique has been discarded. To cope with the relationship extraction process, *LiReWi* requires that the electronic textbook is previously processed in order to extract the domain topics, to which end *LiTeWi* (Conde et al., 2015) can be used. Then, *LiReWi* elicits the pedagogical relationships between the topics that will be used to build the Educational Ontology. This process is shown in Figure 5.2.

To elicit the pedagogical relationships between the domain topics, *LiReWi* follows the procedure shown in Figure 5.3. First, all the topics are mapped to the diverse knowledge bases (e.g. Wikipedia, WordNet and others derived from both) that will be used to identify the relationships. Then, several relationship extractors, each using a different approach, are concurrently run to elicit candidate relationships. Finally, the results are combined and filtered to obtain the final set of pedagogical relationships. In the next subsections, each step is described in more detail.

Again, *LiReWi* has been firstly tested on the *Principles of Object-Oriented Programming* (Wong and Nguyen, 2010) in order to determine its optimal set-up and, then, evaluated on the *Introduction to Astronomy* (Morison, 2008) textbook.

5.2.1 Mapping Topics to Knowledge Base Resources

To extract pedagogical relationships between topics, *LiReWi* uses, in addition to shallow parsing techniques, several knowledge bases such as Wikipedia, WordNet, WikiTaxonomy, WibiTaxonomy and WikiRelations. To this end, it is necessary to map every topic to its corresponding entries in those knowledge bases. The topics identified by *LiTeWi* are already mapped and disambiguated to Wikipedia articles; WikiTaxonomy, WikiRelations and WibiTaxonomy are based on Wikipedia articles. However, to be able to use WordNet, the topics must still be mapped to WordNet entries. WordNet organizes words (nouns, verbs, adjectives and adverbs) into cognitive synonyms called synsets. Each synset refers to a distinct concept that can be referred to using different forms. Navigli and Ponzetto (2012) and Fernando (2013) faced a similar problem and defined the mappings or equivalences between Wikipedia articles and Wordnet synsets.

The procedure shown in Figure 5.4 is applied to identify the corresponding synset for each topic. In the example, the topic to be mapped to WordNet is *syntax*, which is related to *Computer Science*. Aiming at carrying out an efficient mapping process,

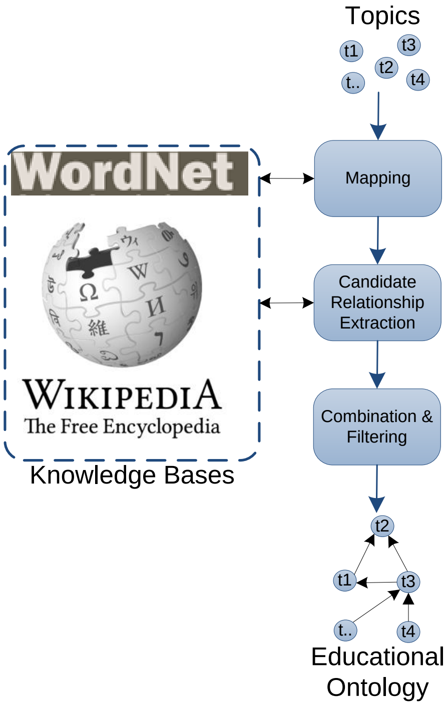
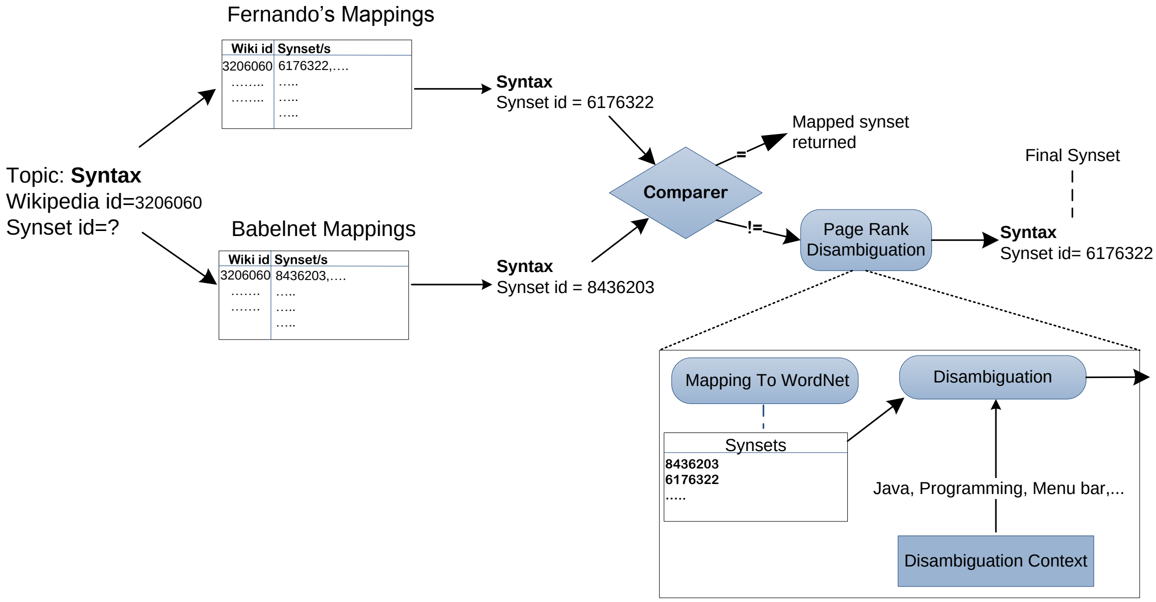


Figure 5.3 – LiReWi Process Overview



Wiki id	Synset/s
3206060	6176322,....
....	.....
....	.....
	.....

Wiki id	Synset/s
3206060	8436203,....
....	.....
....	.....
	.....

Synsets
8436203
6176322
.....

Figure 5.4 – Example of the Algorithm for Topic Mapping in Wikipedia and WordNet

the mapper looks first for the appropriate equivalent synset in those mappings identified in BabelNet Project Navigli and Ponzetto (2012), and also in those mappings discovered by Fernando (2013). If the same synset is found in both cases, the mapper assumes that there are no ambiguity problems and returns the identified synset. Otherwise, a disambiguation process is carried out to identify which of the candidate synsets is the appropriate one. To this end, a Page Rank Mapping Disambiguation step is carried out using UKB (Agirre and Soroa, 2009), a tool for Word Sense Disambiguation and for determining lexical similarity using a pre-existing knowledge base such as Wikipedia or WordNet. UKB requires a context to fulfil its goal. The context is obtained from the topics extracted by *LiTeWi* along with the domain relatedness *LiTeWi* assigned to each of them. The topics with highest domain relatedness score and with a unique meaning in WordNet constitute the context that allows choosing the synset for the topic. In the example of Figure 5.4, the mapped synsets returned by Navigli and Ponzetto (2012) and Fernando (2013) mappings are different. Therefore, the Page Rank Mapping Disambiguation step is carried out to determine the final synset of *syntax* in WordNet. The context used in the example entails topics such as *Programming*, *Menu bar* and *Java*. The Page Rank Mapping Disambiguation mechanism could select a different synset from those proposed by Navigli and Ponzetto (2012) and Fernando (2013).

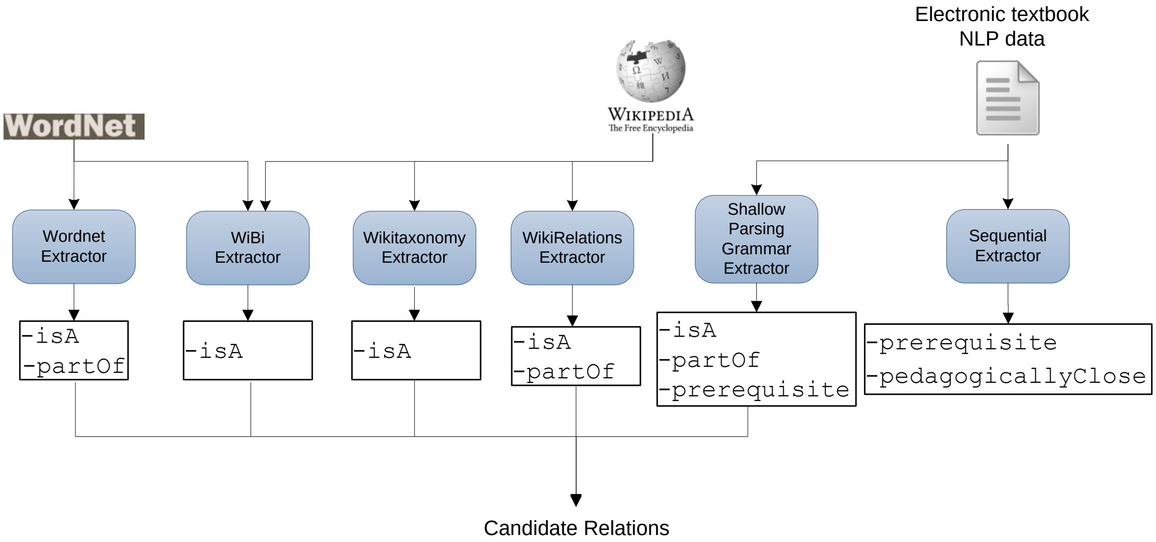


Figure 5.5 – Relationship Extractors used in LiReWi

5.2.2 Relationship Extraction using Different Methods

To extract the pedagogical relationships, *LiReWi* exploits various sources of information and techniques that include taxonomy-based, grammar-based, and co-occurrence-based methods. In Figure 5.5, the Relationship Extractors used in *LiReWi*, along with the type of extracted relationships and the knowledge bases used to this end by each of them, are shown. Each of the extractors identifies a set of candidate relationships along with their confidence, i.e., the trust the extractor has in that relationship being correct. Taxonomy-based methods – WordNet Extractor, WibiTaxonomy Extractor and WikiTaxonomy Extractor– use Formula 5.2 for calculating the confidence while other methods employ their own formula. This information is used in the relationship Combination and Filtering step (see Section 5.2.3). Some of the extractors rely on empirically defined thresholds to fulfil their task. Next, each extractor is described.

5.2.2.1 WordNet Extractor

WordNet (Fellbaum, 1998) can be considered as a huge graph of topics connected by semantic relationships. *LiReWi* uses WordNet to infer relationships from the hypernym relationships (*isA*) and meronym relationships (*partOf*) between the synsets. The processed topics are those identified by *LiTeWi*. The procedure of extracting relationships with WordNet is described next. First, a Deep First Search (DFS) is carried out for each input topic to find the shortest upwards path between the topic and other input topic in WordNet. This search is done in order to gain information from the transitivity of the relationships between the topics. To prevent WordNet from eliciting relationships from paths which are too long, the maximum length of the path can be set. By default, the maximum length is restricted to 3 levels far. This path length threshold has been empirically determined when setting up the system. WordNet Extractor has been tested with different path length thresholds. As it can be observed in Figure 5.6, different path lengths produce different outputs regarding the number of extracted relationships and their correctness. In this case, the selected path length was 3, as it produces balanced output in terms of number of identified relationships, their correctness, and the computational load.

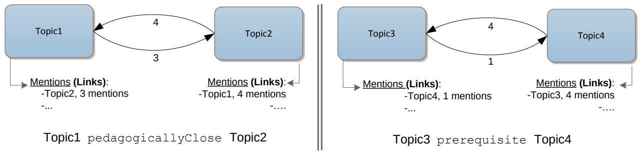


Figure 5.6 – tal and Correct Relationships using different path lengths for WordNet Extractor

Finally, the system determines the confidence of the relationship considering the length of the path. The shorter the path, the greater the confidence of the relationship is. Formula 5.2 is used to calculate the confidence, where b is the base confidence, which is 1 for WordNet Extractor, and p represents the path length.

$$\text{Confidence} = \max(0, b - (0.1 * (p - 1))) \tag{5.2}$$

An example of the application of WordNet Extractor can be seen in Figure 5.7. In the figure, the nodes represent the WordNet synsets that are connected between them via semantic relationships such as hypernym relationships or meronym relationships. The rectangles represent topics that are mapped to WordNet synsets, whereas the circles represent WordNet synsets not mapped to input topics. When a path including only relationships of the same kind between two topics and which is also shorter than the maximum length is found, a pedagogical relationship is defined. In the figure, it can be observed that *Mars* and *Terrestrial Planet* are linked by a hypernym relationship-based path. Therefore, *isA*relationship is inferred between those topics. On the other hand, *Mars* and *Solar System* are related through meronym relationships. In this case, *partOf* is inferred between those topics. The confidence of the *Mars partOf Solar System* relationship is calculated using only the

Table 5.5 – Relationships Extracted by WordNet Extractor

Relationship	**Confidence**
Earth *partOf* Solar system	1
Earth *isA* a Planet	1
Saturn *partOf* Heliosphere	0.9
Sun *partOf* Heliosphere	0.9
Jupiter *partOf* Milky Way	0.8
Saturn *partOf* Milky Way	0.8

base confidence parameter, which is 1 in WordNet Extractor. "Mars *isA* Terrestrial planet" relationship confidence is calculated with a path length of 2 therefore, the resulting confidence is 0.9.

Examples of relationships extracted by the WordNet Extractor with their assigned confidence from the *Introduction to Astronomy* textbook are shown in Table 5.5.

5.2.2.2 WibiTaxonomy Extractor

WibiTaxonomy (Flati et al., 2014) is a knowledge base that comprises two interconnected taxonomies, the Wikipedia article taxonomy and the category taxonomy. Extracting relationships from WibiTaxonomy entails two steps. First, each topic is mapped to the taxonomy of the articles using the mapped Wikipedia article of each topic. Then, each topic is also mapped to the taxonomy of the categories using the parent categories of the topic in Wikipedia. In the second step, using a similar pro-

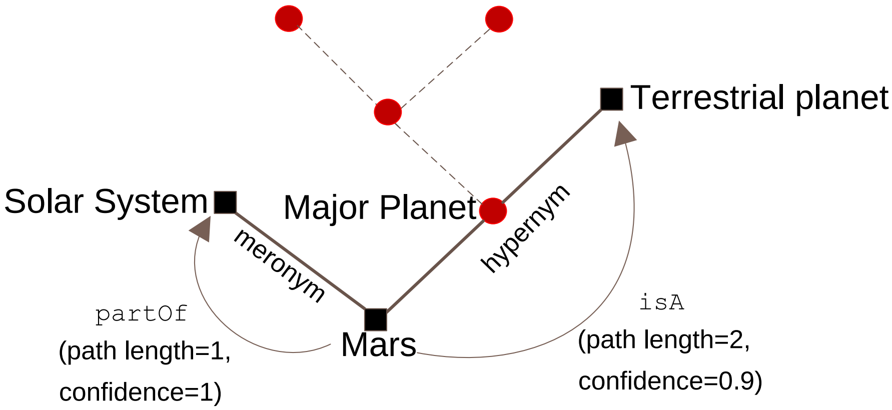


Figure 5.7 – Example of the Application of WordNet Extractor

cedure to that used by the WordNet Extractor, WibiTaxonomy Extractor looks for paths of a limited length to infer the relationships from the taxonomies of both articles and categories. Again, the confidence of each extracted relationship is adjusted using the Formula 5.2 with 0.8 base confidence.

In Figure 5.9, a graphical example of a relationship derived using WibiTaxonomy Extractor can be seen. This example shows that the extractor inferred that "Trapezium Cluster *isA* Open Cluster" with 1-step path and 0.8 confidence.

To determine the optimal maximum path length, some tests were conducted when setting up the system. The results showed that using a path length of at most 3 levels to gain information from the transitivity of the relationships provides the optimal results regarding the total extracted relationships and their correctness. Figure 5.8 summarises the results of these tests.

In Table 5.6, some examples of extracted relationships inferred by WibiTaxonomy Extractor can be seen with their assigned confidence from the *Introduction to Astronomy* textbook.

Table 5.6 – Relationships Extracted by WibiTaxonomy Extractor

Relationship	**Confidence**
Earth's rotation *isA* Rotation	0.8
Trapezium cluster *isA* Open cluster	0.8
Nix (moon) *isA* Natural satellite	0.8
Tau neutrino *isA* Elementary particle	0.8
Trapezium cluster *isA* Star	0.6

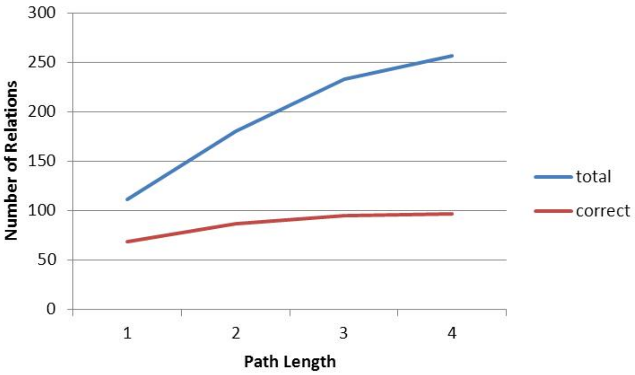


Figure 5.8 – Total and Correct Relationships using different path lengths for WibiTaxonomy Extractor

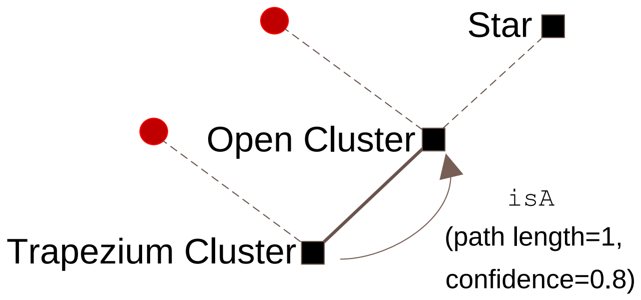


Figure 5.9 – Example of the Application of WibiTaxonomy Extractor

5.2.2.3 WikiTaxonomy Extractor

The WikiTaxonomy (Ponzetto and Strube, 2007) is a huge taxonomy derived from the Wikipedia category system where all the links between categories are represented by *isA* relationships. Moreover, WikiTaxonomy contains a dictionary where the articles are mapped to the corresponding category entries in the taxonomy. The WikiTaxonomy extractor carries out the following procedure to elicit the taxonomic relationships between topics. First, each topic is mapped to its corresponding WikiTaxonomy categories. Then, a DFS is carried out to find the shortest upwards path between the topics considering the categories in the WikiTaxonomy. Once again, the maximum admissible path length was configured. This has been empirically determined making a set of tests like those in WordNet and WibiTaxonomy Extractors. The tests show that the optimal results are obtained using 1 level as limit. Results are shown in Figure 5.10.

Moreover, the confidence on the relationship is likewise computed considering the path distance using Formula 5.2 with the previously empirically determined base confidence (0.8). Figure 5.11 shows an example in which the WikiTaxonomy Extractor identifies the *isA* relationship between *Astronomical Unit* and *Measurement* with 0.5 confidence. In this figure, the squares represent the input topics and the circles represent the WikiTaxonomy network section where a path of length 2 between both topics has been found.

Some examples of extracted relationships using WikiTaxonomy Extractor with their assigned confidence from the *Introduction to Astronomy textbook* are shown in Table 5.7.

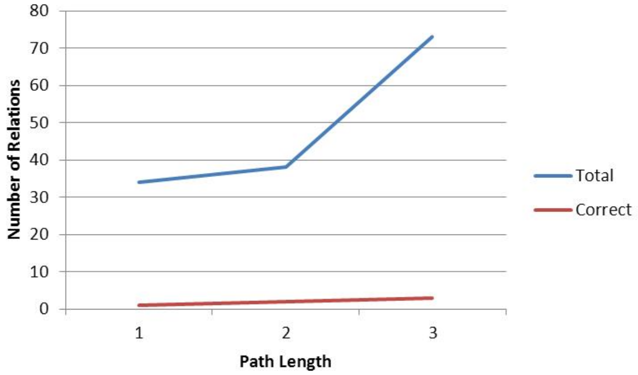


Figure 5.10 – Total and Correct Relationships using different path lengths for WikiTaxonomy Extractor

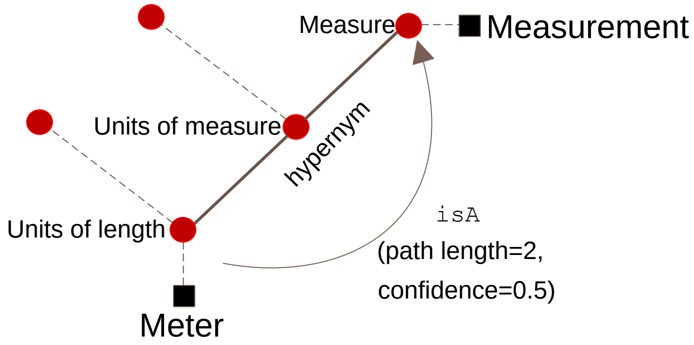


Figure 5.11 – Example of the Application of WikiTaxonomy Extractor

5.2.2.4 WikiRelations Extractor

The WikiRelations (Nastase and Strube, 2008) knowledge base comprises a big set of tuples between Wikipedia categories containing several kinds of relationships. In this work, only the subset of tuples containing *isA* or *partOf* relationships has been employed. The WikiRelations Extractor carries out the procedure shown in Figure 5.12. First, for each topic it gets its corresponding Wikipedia article to extract parent categories associated with that article and map them to one or more WikiRelations tuples. In the example, *Light* and *Electromagnetic radiation* are each associated with categories that appear in two WikiRelations tuples. Then, it filters the tuples where topics are mapped. Whenever a tuple contains two of the input topics, a relationship between those topics is inferred.

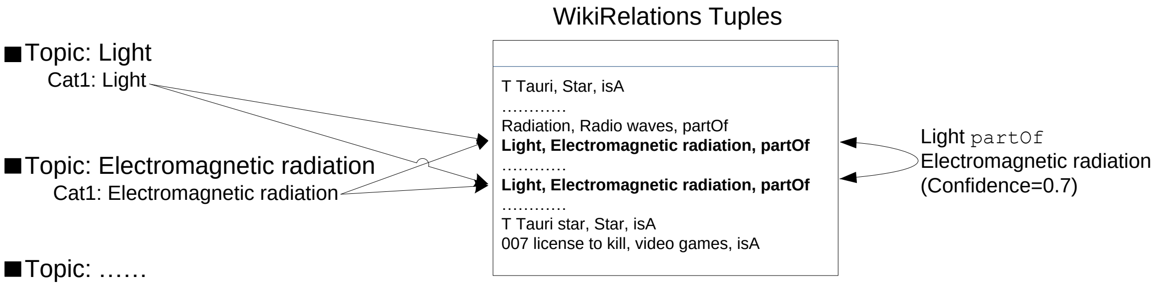

WikiRelations Tuples
■ Topic: Light
Cat1: Light
■ Topic: Electromagnetic radiation
Cat1: Electromagnetic radiation
■ Topic: ……
T Tauri, Star, isA
…………
Radiation, Radio waves, partOf
Light, Electromagnetic radiation, partOf
…………
Light, Electromagnetic radiation, partOf
…………
T Tauri star, Star, isA
007 license to kill, video games, isA
Light partOf
Electromagnetic radiation
(Confidence=0.7)

Figure 5.12 – Example of the Application of WikiRelations Extractor

As tuples containing the same categories can be found more than once in WikiRelations, this fact is considered to calculate the confidence of each relationship. In the example, it can be seen that WikiRelations have inferred two times a *partOf* relation. So, the confidence of the relationship is adjusted accordingly using Formula 5.3, where b represents the base confidence (0.6), and n being the number of tuples.

$$\text{Confidence} = \min(b + (0.1 * n), 1) \tag{5.3}$$

A test has been done to establish the confidence threshold. In this case the threshold filters out those relationships that have been inferred only once (those that have only one tuple in WikiRelations). The results of this test can be seen in Figure 5.13. Again, higher *precision* has been prioritized over high *recall* while empirically determining the threshold value.

In Table 5.8 some examples of relationships with their corresponding confidence are shown from the *Introduction to Astronomy* textbook.

5.2.2.5 Shallow Parsing Grammar Extractor

The Shallow Parsing Grammar Extractor can infer *isA*, *partOf* and *prerequisite* relationships applying a grammar on the part-of-speech information of the input text-

Table 5.7 – Relationships Extracted by WikiTaxonomy Extractor

Relationship	**Confidence**
Equinox *isA* Astrometry	0.6
Optical aberration *isA* Optics	0.6
Electron *isA* Matter	0.5
Hydrogen *isA* Matter	0.5
International Atomic Time *isA* Measurement	0.7

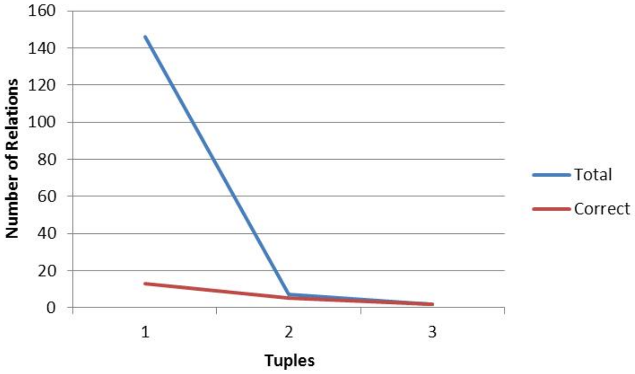


Figure 5.13 – Total and Correct Relationships using different path lengths for WikiRelations Extractor

Table 5.8 – Relationships Extracted by WikiRelations Extractor

Relationship	**Confidence**
Lithium *isA* Metal	1
Microwave *partOf* Electromagnetic spectrum	1
Earth *partOf* Solar system	1
Alpha Centaur *partOf* Centaury	1
Solar Wind *partOf* Solar system	0.8

book. Larrañaga et al. (2014) defined a grammar for the extraction of pedagogical relationships applied to the Basque language. This grammar is applied to morphological information using the CG3 parser[1]. In this thesis, a similar grammar has been developed for English. The grammar consists of a set of rules that are triggered when the corresponding pattern is met. Some of those patterns are shown in Table 5.9. Next, the process followed by *LiReWi* to extract relationships using the Shallow Parsing Grammar is described (see Figure 5.14). First, the extractor identifies those sentences in which the input topics are referred. In addition, the topics being referred are annotated with the part-of-speech information of the sentence. As some of the input topics might subsume others, e.g., *sun-eclipse* - *eclipse*, the system resolves this situation considering a simple matching algorithm where those compound terms have prevalence over the simple ones. The sentences containing more than one mention of input topics will be selected as they may suggest a relationship between the involved topics. Next, the shallow parsing grammar is applied to the sentences

[1] http://beta.visl.sdu.dk/constraint_grammar.html

Table 5.9 – Examples of Patterns for Relationships Extraction

Pattern	**Example**
TOPIC +called|referred to as + TOPIC	Scientists believe that the galaxy referred to as Milky Way has over 100 billion stars.
TOPIC + TO BE + [det] + TOPIC	Earth is a planet.
TOPIC + consist + of + TOPIC	Galaxy consists of stars.
TOPIC + to be component(s) + of +[det] + TOPIC	Galaxies are the main components of the universe.
TOPIC + of + [det] + TOPIC	The movements of the planets.

extracted in the previous step. Finally, taking into account the information of each triggered rule, specifically, the type of relationship, the direction of the relationship and the topics that triggered the rule, a relationship is inferred between those topics also obtaining the confidence of the triggered rule. The confidence of each rule was previously determined from its precision after testing it with a set of examples applied to the *Principles of Object-Oriented Programming* (Wong and Nguyen, 2010) textbook.

5.2.2.6 Sequential Extractor

This extractor aims to elicit sequential relationships such as *prerequisite* and *pedagogicallyClose*. The Sequential Extractor uses the information contained in the processed textbook along with information gathered from Wikipedia to extract these kinds of relationships. In particular, it uses the co-occurrences of the topics within the sentences along with the Wikipedia link structure between articles. To use the informa-

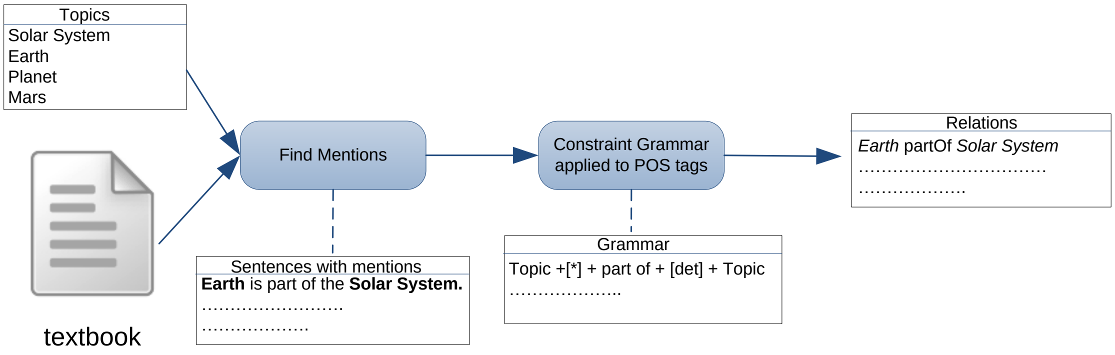

Topics
Solar System
Earth
Planet
Mars
textbook
Find Mentions
Sentences with mentions
Earth is part of the **Solar System.**
………………………
…………………
Constraint Grammar applied to POS tags
Grammar
Topic +[*] + part of + [det] + Topic
………………..
Relations
Earth partOf *Solar System*
……………………………
……………….

Figure 5.14 – Example of the Application of the Shallow Parsing Grammar Extractor

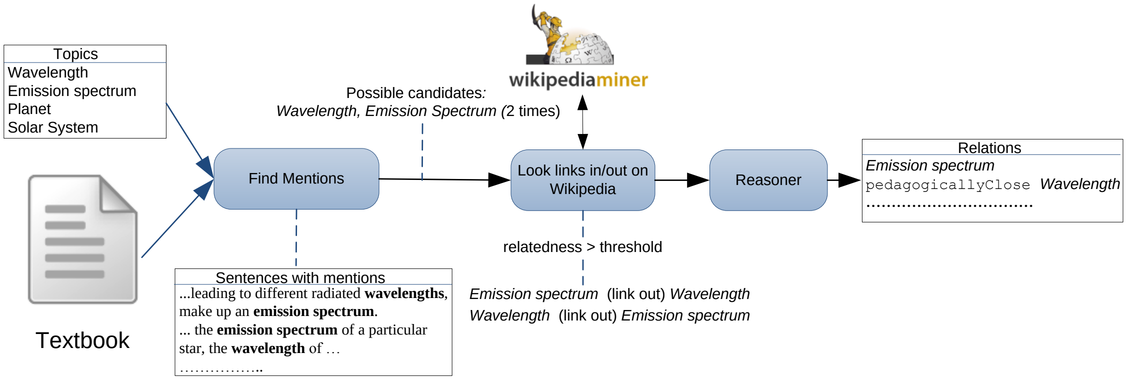

Topics
Wavelength
Emission spectrum
Planet
Solar System
Textbook
Find Mentions
Possible candidates:
Wavelength, Emission Spectrum (2 times)
wikipediaminer
Look links in/out on Wikipedia
Reasoner
Relations
Emission spectrum pedagogicallyClose Wavelength
....
relatedness > threshold
Emission spectrum (link out) Wavelength
Wavelength (link out) Emission spectrum
Sentences with mentions
...leading to different radiated wavelengths, make up an emission spectrum.
... the emission spectrum of a particular star, the wavelength of …
……………..

Figure 5.15 – Example of the Application of the Sequential Extractor

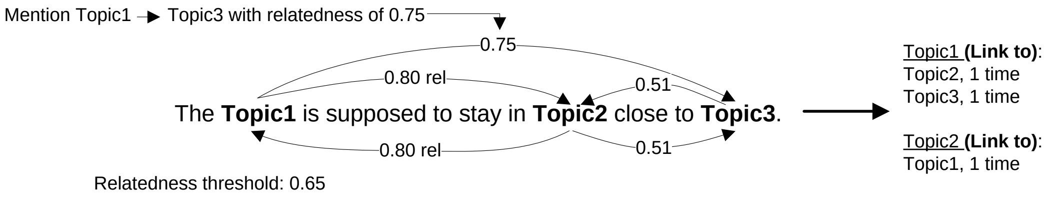

Mention Topic1 → Topic3 with relatedness of 0.75
0.75
0.80 rel
0.51
The Topic1 is supposed to stay in Topic2 close to Topic3.
0.80 rel
0.51
Relatedness threshold: 0.65
Topic1 (Link to):
Topic2, 1 time
Topic3, 1 time
Topic2 (Link to):
Topic1, 1 time

Figure 5.16 – Example of Getting Mentions from Sentences

tion of the link structure between articles, this module uses WikiMiner (Milne and Witten, 2013). Next, the procedure is described (see Figure 5.15).

First, as occurs in the Shallow Parsing Grammar Extractor, the extractor identifies the topics that are being referred in the text. Once again, the system applies a simple matching algorithm where the compound terms have prevalence over the simple ones. The output of this process is a list of sentences that contain mentions of the input topics. Next, for each of those sentences, a reference relationship is defined between each pair of topics appearing in the sentence if the first topic refers to the second. A topic is considered to refer to another if a link out from the first topic to the second exists in Wikipedia with a relatedness score beyond an empirically gathered threshold. *LiReWi* uses WikiMiner to compute the relatedness score of two topics. For example (see Figure 5.16) *topic1* and *topic2*, which have links in both directions in Wikipedia, appear in the same sentence. As their relatedness is higher than the empirically determined threshold (0.7) a link between them is annotated. *Topic3* only references *Topic2*, but their relatedness is below the threshold.

$$\text{Confidence} = \min(b + (top1m + top2m - low * 0.05), 1) \tag{5.4}$$

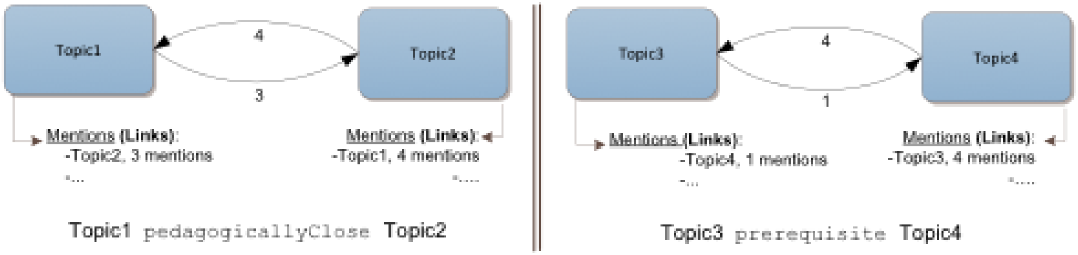


Figure 5.17 – Examples of the Application of the Sequential Extractor

Finally, for each linked topic pair, a sequential relationship is inferred. If the links between both topics are balanced, i.e., the number of links from the first topic to the second is similar to the number of links from the second to the first, a *pedagogicallyClose* relationship between both topics is inferred. Otherwise, a *prerequisite* relationship is inferred from the topic with the highest number of outgoing links to the topic with higher incoming links. Figure 5.17 shows two examples in which a *pedagogicallyClose* and a *prerequisite* relationships are inferred using this procedure. The confidence of the extracted relationships is calculated using the Formula 5.4, where b is the base confidence (0.6), $top1m$ is the number of links from the first topic, $top2m$ is the number of links from the second topic and low is the threshold determining the minimum number of links for a relationship to be inferred, 2 in this case.

Some examples of extracted relationships using this extractor are shown in Table 5.10 with their assigned confidence.

Table 5.10 – Relationships Extracted by Sequential Extractor

Relationship	**Confidence**
Emission spectrum *pedagogicallyClose* Wavelength	1.0
Wavelength *pedagogicallyClose* Emission spectrum	0.9
Helium *pedagogicallyClose* Atom	0.9
Radiation pressure *prerequisite*Comet tail	0.6
Space *prerequisite* Planet	0.6

5.2.3 Combining and Filtering Relationships

In the last phase, following a three-step process, *LiReWi* obtains the final set of relationships from the relationship candidates obtained using the extractors described

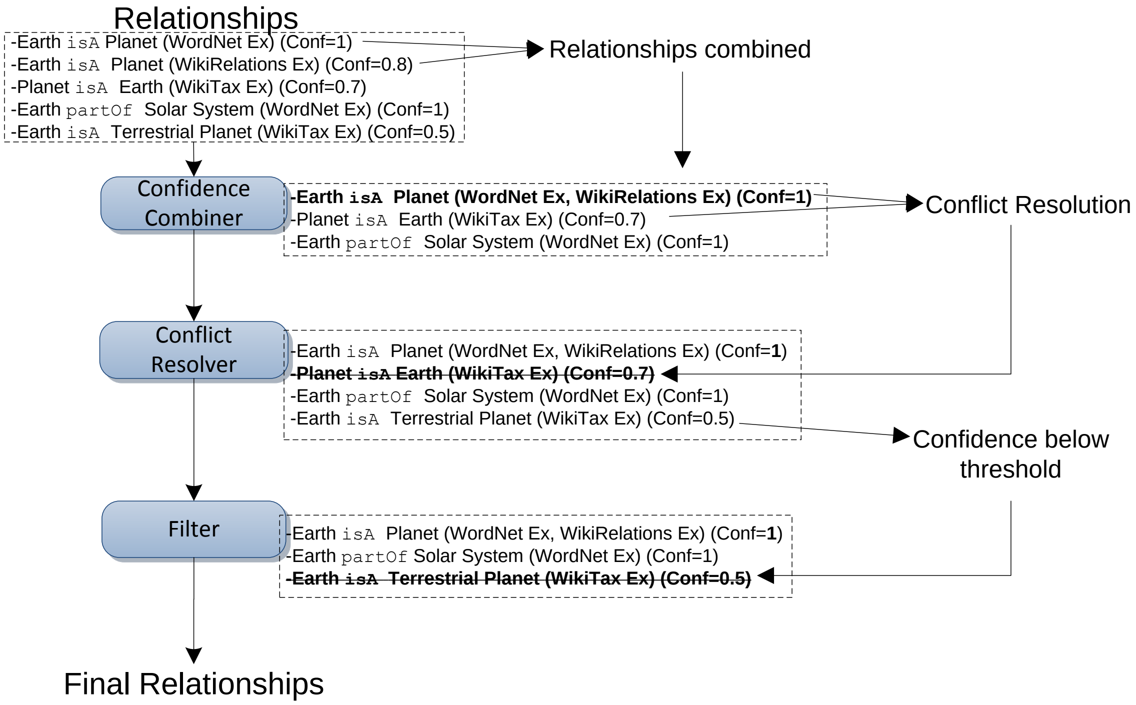

Relationships
-Earth isA Planet (WordNet Ex) (Conf=1)
-Earth isA Planet (WikiRelations Ex) (Conf=0.8)
-Planet isA Earth (WikiTax Ex) (Conf=0.7)
-Earth partOf Solar System (WordNet Ex) (Conf=1)
-Earth isA Terrestrial Planet (WikiTax Ex) (Conf=0.5)
Relationships combined
Confidence Combiner
-Earth isA Planet (WordNet Ex, WikiRelations Ex) (Conf=1)
-Planet isA Earth (WikiTax Ex) (Conf=0.7)
-Earth partOf Solar System (WordNet Ex) (Conf=1)
Conflict Resolution
Conflict Resolver
-Earth isA Planet (WordNet Ex, WikiRelations Ex) (Conf=1)
~~-Planet isA Earth (WikiTax Ex) (Conf=0.7)~~
-Earth partOf Solar System (WordNet Ex) (Conf=1)
-Earth isA Terrestrial Planet (WikiTax Ex) (Conf=0.5)
Confidence below threshold
Filter
-Earth isA Planet (WordNet Ex, WikiRelations Ex) (Conf=1)
-Earth partOf Solar System (WordNet Ex) (Conf=1)
~~-Earth isA Terrestrial Planet (WikiTax Ex) (Conf=0.5)~~
Final Relationships

Figure 5.18 – Example of the Combination and Filtering Processes

above. It starts by combining and adjusting the confidence of those relationships inferred by more than one extractor, to which end Formula 5.5 is used, where ci is the confidence of extractor i, n is the number of extractors that identified the relationship, and α is a constant (1.1) that promotes relationships identified by several extractors. The more extractors infer a relationship, the higher the confidence in that relationship is. Next, *LiReWi* detects and solves conflicts between relationship candidates of the same kind. In this step, relationships with inconsistencies and erroneous relationships are removed. For example, when a relationship has the same topic as source and destination, it is removed. Furthermore, some relationships may form a so-called loop. For example, one relationship involving two topics may be inferred in both directions. In those cases, *LiReWi* carries out a solving process selecting the final relationships using the confidence as a criterion and the link structure in Wikipedia. In the final step, those relationships that have a confidence below an empirically gathered threshold (0.6) are deleted to improve the consistency of the generated LDO. When two different relationships are identified between two topics, say *isA* and *partOf* the assertion with highest confidence is accepted.

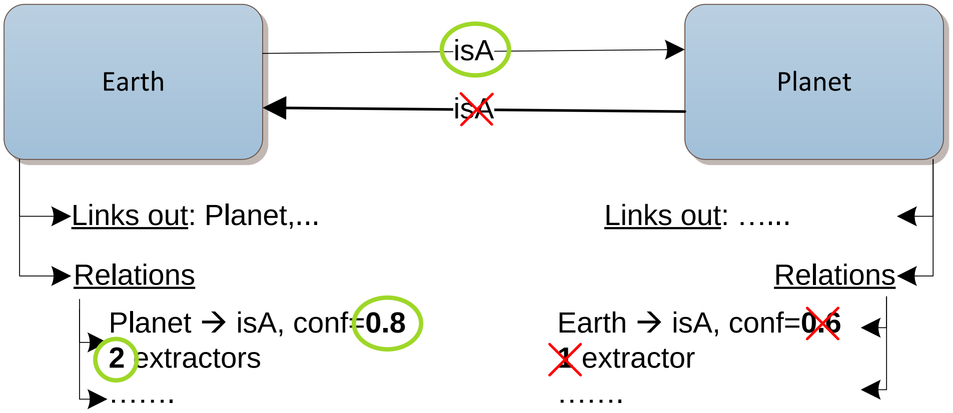


Figure 5.19 – Example of a Conflict Resolution

$$\text{Final Confidence} = \min(\frac{1}{n}\sum_{i=1}^{n} c_i * \alpha, 1) \tag{5.5}$$

Figure 5.18 illustrates the process described above. Firstly, the confidences of the relationships elicited by two or more extractors are combined. For example, "Earth *isA* Planet" is combined and adjusted accordingly.

In the second step, a conflict is found between "Earth *isA* Planet" and "Planet *isA* Earth" proposals. The system looks at the link structure of the topics in Wikipedia, along with the confidence of the extracted relationships, to determine the final relationship. In the figure, "Earth *isA* Planet" has higher confidence than "Planet *isA* Earth". In addition, *Earth* has a link to *Planet* in Wikipedia, whereas *Planet* does not have a link to *Earth*. Therefore, the system decides to discard "Planet *isA* Earth" (see Figure 5.19). Finally, the system deletes those relationships that have less confidence than the predetermined threshold. In this case, "Earth *isA* Terrestrial planet" is deleted because its confidence is lower than the threshold.

5.3 Evaluation

In this section, the experiment conducted to evaluate *LiReWi* is depicted. *LiReWi* has been evaluated using two approaches, *Gold-standard* and expert validation.

This time, *LiReWi* has firstly been tuned up on the *Principles of Object-Oriented Programming* (Wong and Nguyen, 2010) in order to determine its optimal set-up, and subsequently, evaluated on the *Introduction to Astronomy* (Morison, 2008) textbook.

First, an evaluation of the mapping techniques is depicted. Then, the evaluation of the candidate relationship extraction is presented and, finally, the evaluation of the combination and filtering is described.

LiReWi requires a set of topics as input. Therefore, the *Introduction to Astronomy* textbook has been firstly processed with *LiTeWi* (Conde et al., 2015) to obtain such a topic set. Next, the topics with highest relatedness with the domain of the textbook have been selected and used as input for the relationship elicitation. In this experiment the input set entailed 199 topics. The relatedness value used for this purpose was the CValue (Frantzi et al., 2000) score computed by *LiTeWi* for the extracted topics.

As mentioned above, the evaluation procedure conducted combined *Gold-standard* and expert validation to measure the performance of the system. For the *Gold-standard* evaluation, four experts stated the set of gold relationships (*partOf*, *prerequisite pedagogicallyClose* and *isA*) between the 199 input topics. The *Gold-standard* entails 174 relationships, being 15 *pedagogicallyClose*, 10 *prerequisite*, 69 *partOf* and 80 *isA*. Then, the results obtained by the different extractors have been compared with the *Gold-standard.* Regarding the expert validation, once again 4 experts have manually checked the correctness of the extracted relationships. Fleiss's kappa (Fleiss, 1971) coefficient has been computed to measure the inter-rater agreement. The experts agreed on 270 of 295 total extracted relationships, with 0.974 weighted kappa score. This value shows an almost perfect agreement between the experts (Landis and Koch, 1977). The results on the validation will be on those relationships agreed on by all the experts.

5.3.1 Results of the Mapping

Table 5.11 shows the results of the evaluation of the mapping step in the *Introduction to Astronomy.* BabelNet approach led to the highest *precision* 100%, but its *recall* was the lowest with only 14.73%. Fernando's method, on the other hand, led to 83.33% *precision* with 18.42% *recall.* Our approach, which combines both methods with UKB, results in 97.82% *precision* and 23.68% *recall*, showing that it greatly increases the *recall* while minimizing the loss on *precision.* The *F1-score* is also shown in the table.

5.3.2 Results of the Candidate Relationship Extraction

In this section the performance of each extractor is depicted. For each extractor, the performance is reported by comparing the relationships it has extracted against

Table 5.11 – WordNet/Wikipedia Mapping Results

	Precision(%)	**Recall(%)**	**F1-Score(%)**
Babelnet	**100**	14.73	25.68
Fernando	83.33	18.42	30.17
Page Rank	97.82	**23.68**	**38.13**

the *Gold-standard* (*precision*, *recall* and *F1-score*). In addition, the expert validation results, i.e., the percentages of correct relationships according to the experts (*correctness*) are included. The performance of the extractors is summarised in Table 5.12. The WordNet Extractor identified 35 relations achieving 77.14% *recall* with 15.51% *precision.* The expert validation resulted in 100% of the identified relations being valid. The WikiTaxonomy Extractor extracted 45 relations from the selected topics. The extractor obtained 8.88% *precision* and 2.29% *recall* for *Gold-standard* validation. The expert validation showed that only 4 of them (8.88%) were valid. The WikiRelations Extractor identified 26 relations, obtaining 69.23% *precision*, with 10.34% *recall.* The expert validation resulted in being 20 correct (76.92%). The WibiTaxonomy Extractor identified 138 relations achieving 39.85% *precision* with 31.6% *recall* for the textbook. The expert validation shows that 70 (50.72%) of the identified relations were valid. The Shallow Parsing Grammar Extractor identified 11 relations, none of them being part of the *Gold-standard.* The expert validation determined that 4 (36.36%) of the identified relations were valid. Finally, the Sequential Extractor identified 15 relationships. This method achieved 53.33% *precision* and 4.59% *recall* considering the *Gold-standard.* The expert validation determined that 9 of them (60%) were correct.

Table 5.12 – Results of the Candidate Extraction by Extractor

			Gold Standard		**Exp. Validation**
	Num. Rel.	**Precision (%)**	**Recall (%)**	**F1-Score (%)**	**Correctness (%)**
WordNet	35	**77.14**	15.51	25.83	**100**
WikiTaxonomy	45	8.88	2.29	3.64	8.88
WikiRelations	26	69.23	10.34	17.99	76.92
WibiTaxonomy	**138**	39.85	**31.6**	**35.25**	50.72
Shallow Parsing	11	0	0	0	36.36
Sequential	15	53.33	4.59	8.45	60

The WordNet Extractor shows the best performance in terms of the expert validation. This result was quite predictable taking into account that WordNet contains manually defined relationships. However, WordNet may be currently limited in terms of *recall* as it is not being actively updated. The extractors based on Wikipedia (WikiTaxonomy, WikiRelations and WibiTaxonomy) showed diverse behaviour; the newer the underlying method, the better the results are. WikiTaxonomy Extractor showed the best performance among those extractors based on Wikipedia. The Shallow Parsing Extractor did not extract any relations considered for the *Gold-standard*. However, the expert validation shows that it extracted valuable relations from the selected topics. The Sequential Extractor achieved remarkable results.

5.3.3 Results of the Combination and Filtering

In this section the overall results of *LiReWi* are depicted. Once all the extractors were processed in parallel on the *Introduction to Astronomy* textbook, their results were combined and filtered as described in Section 5.2.3. The performance of the Combination and Filtering step is presented in Table 5.13. 266 different relations were inferred by *LiReWi* using all the extractors. Considering the *Gold-standard*, 36.21% *precision* and 50.57% *recall* were achieved. The experts considered that 117 of the 266 (43.98%) identified relationships were correct.

Table 5.13 – LiReWi Performance in the “Introduction to Astronomy” Textbook

		Gold Standard		**Exp. Validation**
	Precision(%)	**Recall(%)**	**F1-Score(%)**	**Correctess (%)**
LiReWi	36.21	50.57	42.2	43.98

Next, the results for each kind of relation are depicted (see Table 5.14). *LiReWi* extracted a total of 213 *isA* relationships achieving 30.38% *precision* with 76.25% *recall*. The expert validation resulted in 87 of the identified relations being valid (40.84%). 37 *partOf* relationships from the selected topics have been inferred by *LiReWi*, obtaining 51.34% *precision* and 27.54% *recall* for *Gold-standard* validation. The expert validation shows that 19 of them were valid (51.36%). Regarding the *prerequisite* relationships, 10 have been extracted, obtaining 30% *precision* and with 30% *recall*. The expert validation resulted in 8 of them being correct (80%). Finally, *LiReWi* extracted 6 *pedagogicallyClose* relationships achieving 50% *precision* with 33% *recall* for the textbook. The expert validation shows that 3 of the identified relations were valid (50%).

Table 5.14 – LiReWi Extractors Performance

		Gold-standard			**Exp. Validation**
	Rels.	**Precision**	**Recall**	**F1-Score**	**Correctness**
		(%)	(%)	(%)	(%)
isA	213	30.38	76.25	43.44	40.84
partOf	37	51.34	27.54	35.85	51.36
prerequisite	10	30	30	30	80
pedagogicallyClose	6	50	33	39.75	50

In comparison with each extractor (see Figure 5.20), *LiReWi* outperforms the best extractor (WibiTaxonomy Extractor) by more than 20% in terms of *recall*. Taking into account the *F1-score*, *LiReWi* outperforms the best extractor (WibiTaxonomy Extractor) by 10%.

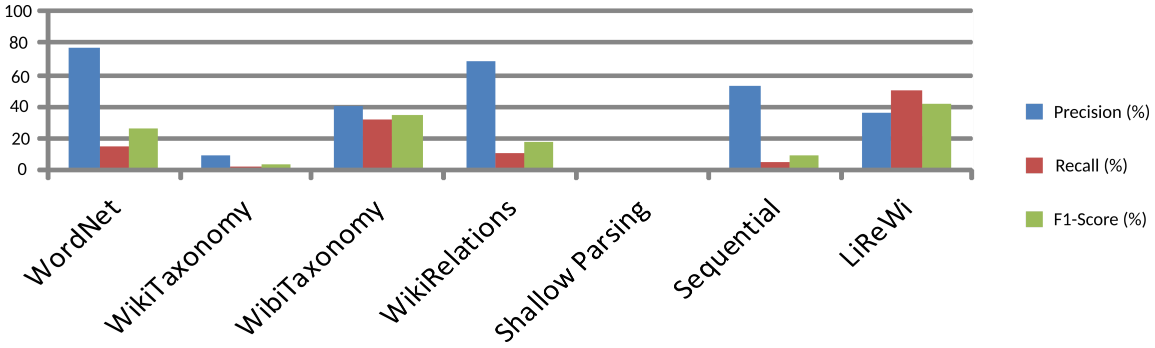


Figure 5.20 – LiReWi vs. Extractors in the Gold-standard Evaluation

The result of the overall process shows that LiReWi can take advantage of different methods in order to extract pedagogical relationships from topics.

5.4 Comparison with Other Systems

In the last few years, several efforts have been made in the elicitation of relationships in Ontology Learning in general, but most of them focused on the elicitation of taxonomic relationships (see Section 3.3). For example, Wang et al. (2015) present an approach for the concept hierarchy extraction from textbooks that uses the document structure and Wikipedia.

Roy (2006) exposes the automatic extraction of pedagogic metadata for document understanding. As regards the pedagogical relationship, she deals with the identification of *prerequisite* concepts to understand the document. To identify these *prerequisite* concepts, she works at sentence level and uses a shallow parsing approach to identify the *defined concept list* –concept defined/explained in the sentence– and the *used concept list* –concepts used to define/explain the defined concept. The *defined concept* will constitute the learning outcome and all the remaining noun phrases in the document, i.e., the *used concept list*, will be considered as *prerequisite* to understand the document. Any concept included in the *used concept list* listed down in a *defined concept list* will be removed from the *used concept list* and, as a consequence, not considered as a *prerequisite* to understand the document.

Liang et al. (2015) proposed a metric to determine measure *prerequisite* relationships between pairs of concepts. To determine if a *prerequisite* exists between two concepts A and B, the metric computes the difference of the weighted references from the concepts related to concept A to topic B and the references from topics related to concept B to concept A. If the score goes beyond a threshold, a *prerequisite* relationship exists. The authors used Wikipedia to look for the references and determine the weights of the concepts. They reported an average of 61% accuracy. The metric can be used by supervised Machine Learning algorithms to identify *prerequisite* relationships.

To our knowledge, *DOM-Sortze* is the only system that addresses the extraction of a set of pedagogical relationships such as *isA partOf*, *next* and *prerequisite*. *DOM-Sortze* reported 63.27% *precision* with 20.74% *recall* in the elicitation of pedagogical relationships (Larrañaga et al., 2014). As can be observed in Table 5.15, *LiReWi* outperforms *DOM-Sortze* considering *recall* and *F1-Score*[2].

Table 5.15 – LiReWi vs. Dom-Sortze Gold-standard Reported Performance

	Precision(%)	**Recall(%)**	**F1-Score(%)**
LiReWi	36.21	**50.57**	**42.2**
DOM-sortze	63.27	20.74	31.24

5.5 Summary

Throughout this chapter, the elicitation of pedagogical relatioships, namely *isA*, *partOf*, *next*, *prerequisite*, and *pedagogicallyClose*, has been described. This chapter

[2] No expert validation was carried out for *DOM-Sortze* regarding the elicitation of pedagogical relationships

first delved into the language-independence of the proposal of *DOM-Sortze* regarding the elicitation of structural relationships (*isA* and *partOf*) from document outlines. To this end, the heuristics were adapted and tested on English outlines. Given that the absence of domain-specific knowledge limited the performance of the approach, it has been improved with the use of Wikipedia, which covers a huge amount of domains. Its inclusion has significantly improved the acquisition of structural relationships, and, therefore, proved the added value of Wikipedia for such a task.

In addition, this chapter also presented *LiReWi*, a pedagogical relationship extractor that combines diverse techniques and resources such as Wikipedia and Wordnet for the elicitation of pedagogical relationships (*isA*, *partOf*, *prerequisite* and *pedagogicallyClose*) from electronic document bodies. To elicit the pedagogical relationships between the domain topics, *LiReWi* first maps all the topics, previously extracted by *LiTeWi*, to the additional knowledge bases used, in this case WordNet. Then, several relationship extractors, each using a different approach, are concurrently run to elicit candidate relationships. Finally, the results are combined and filtered to obtain the final set of pedagogical relationships.

To assess its performance, *LiReWi* has been firstly tested on the *Principles of Object-Oriented Programming* (Wong and Nguyen, 2010) textbook in order to determine its optimal set-up and, then, evaluated on the *Introduction to Astronomy* (Morison, 2008) textbook. The promising results of this evaluation have been also presented throughout this chapter.

The following chapter focuses on the extraction of multilingual LOs.

6

Extraction of Learning Objects

Digital contents used in TSLSs with learning purposes are often referred to as DRs, or LOs if they are annotated with metadata. One of the main advantages of LOs is that they can be reused to support learning in different platforms or contexts.

In this chapter, the automatic identification and extraction of LOs from documents written in English is addressed to confirm the language independence of the approach proposed in *DOM-Sortze*; it was firstly developed to process textbooks written in the Basque language. This chapter starts by describing the process carried out by *DOM-Sortze* to gather LOs (Section 6.1). Then, the changes needed in *DOM-Sortze* to be able to deal with the English language along with the experiment carried out to validate the proposal are described (Section 6.2). Next, the possibility of incorporating Wikipedia and WordNet as knowledge sources in the LO elicitation process is considered (Section 6.3). Then, *LiLoWi*, a multilingual LO extractor that uses Wikipedia and WordNet, is described (Section 6.4). The experiment carried out to evaluate *LiLoWi* is presented next (Section 6.5). Following, a comparison with other approaches is depicted (Section 6.6) and it concludes with some final remarks (Section 6.7).

6.1 Gathering Learning Objects from Documents

In *DOM-Sortze*, the generation of LOs for the domain topics is achieved by identifying and gathering DRs, i.e., consistent fragments of the document related to one or more topics with a particular educational purpose, and dealing with the appropriate metadata. The identification and extraction of these text pieces is carried out in an ontology-driven process that uses NLP techniques. As the LO generating approach presented in *DOM-Sortze* aims to be domain-independent, the only domain-specific knowledge used is the LDO previously gathered from the electronic document. A DR will refer to a fragment of the document meant to be used in the learning sessions (e.g., *definition*, *exercise*, ...) while a LO refers to a reusable DR enriched with metadata. The LO generation process here described was designed for *ErauzOnt* (Larrañaga et al., 2011; Larrañaga et al., 2012), which is part of the *DOM-Sortze* framework. *ErauzOnt* is able to extract the following kinds of LOs from documents written in Basque:

- **Definition**: a passage that explains the meaning of a term.
- **Example**: something characteristic of its kind or that illustrates something being described.
- **Fact**: a piece of information used as evidence or as part of a report or news article.
- **Theory**: a supposition or a collection of ideas intended to explain something.
- **Principle statement**: description of how or why some phenomenon happens.
- **Problem statement**: a concise description of the problem that the students have to address.

These are the kinds of DRs for which syntactic patterns that allow their identification were found (Larrañaga et al., 2011; Larrañaga et al., 2012).

Figure 6.1 describes the process for gathering the LOs from the electronic document, which entails the following tasks: generating DRs from the document, annotating the DRs to become LOs, and, finally, storing the generated LOs in a LOR for further use. The LDO, a DR Grammar, Discourse Markers and a Didactic Ontology (Leidig, 2001) are used to gather DRs from the internal representation of the electronic textbook labeled with the *part-of-speech* information. The LDO, and the ALOCOM ontology (Verbert et al., 2005) are used to build the LOs from the gathered DRs, and, finally, the LOs are stored in the LOR to facilitate their use and reuse.

Next, these steps are described in more detail.

6.1.1 Generation of Didactic Resources

The identification of the DRs is carried out by identifying relevant text fragments that correspond to *definitions*, *examples*, *facts*, *theories*, *principle statements*, and *problem*

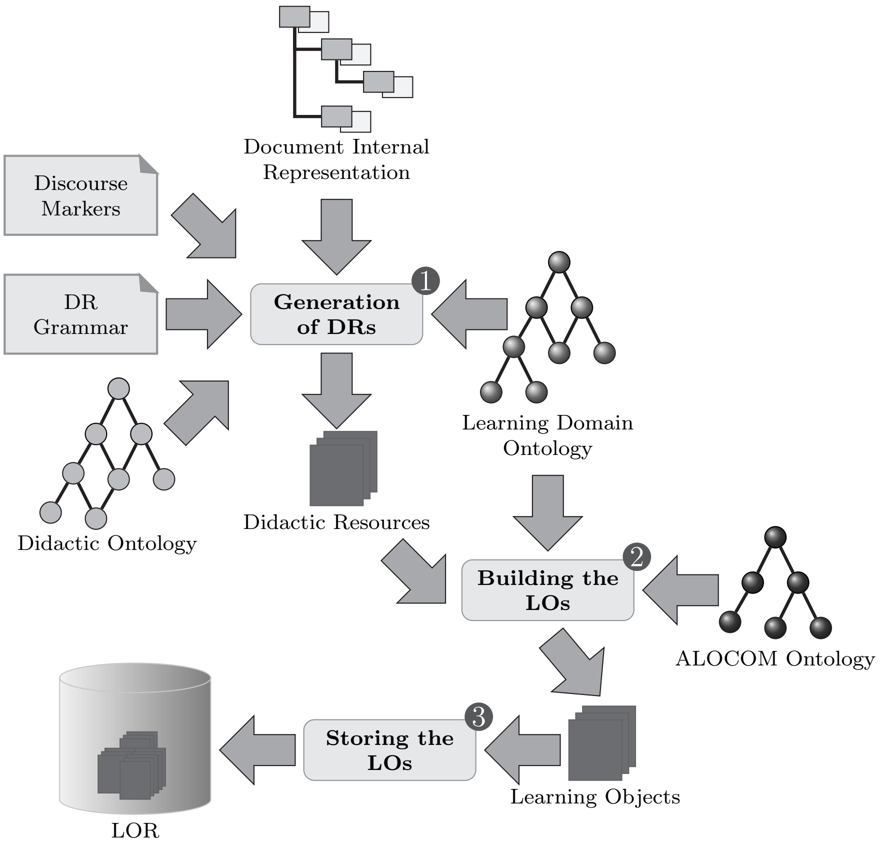


Figure 6.1 – Generation of Learning Objects from Documents (from Larrañaga, 2012)

statements for the LDO topics, as shown in Figure 6.2. First, the appearances of the LDO topics are labelled in the internal representation of the document with the *part-of-speech* information. Next, the DR Grammar is used to find text fragments that might contain appropriate resources. The DR Grammar includes a set of rules that define the different patterns (DR rules) (Larrañaga et al., 2012). These patterns are the most common syntactic structures observed in several topic *definitions*, *examples*, and so on. The grammar for gathering the DRs from the electronic document has been developed using the *Constraint Grammar* formalism.

The identified DRs contain the sentence that triggered the rule for the corresponding DR and all the sentences that follow which refer to the same topic(s). Every DR is labelled with the domain topics referred and with the DR rules that identified it. This information is used later in the LO annotation process.

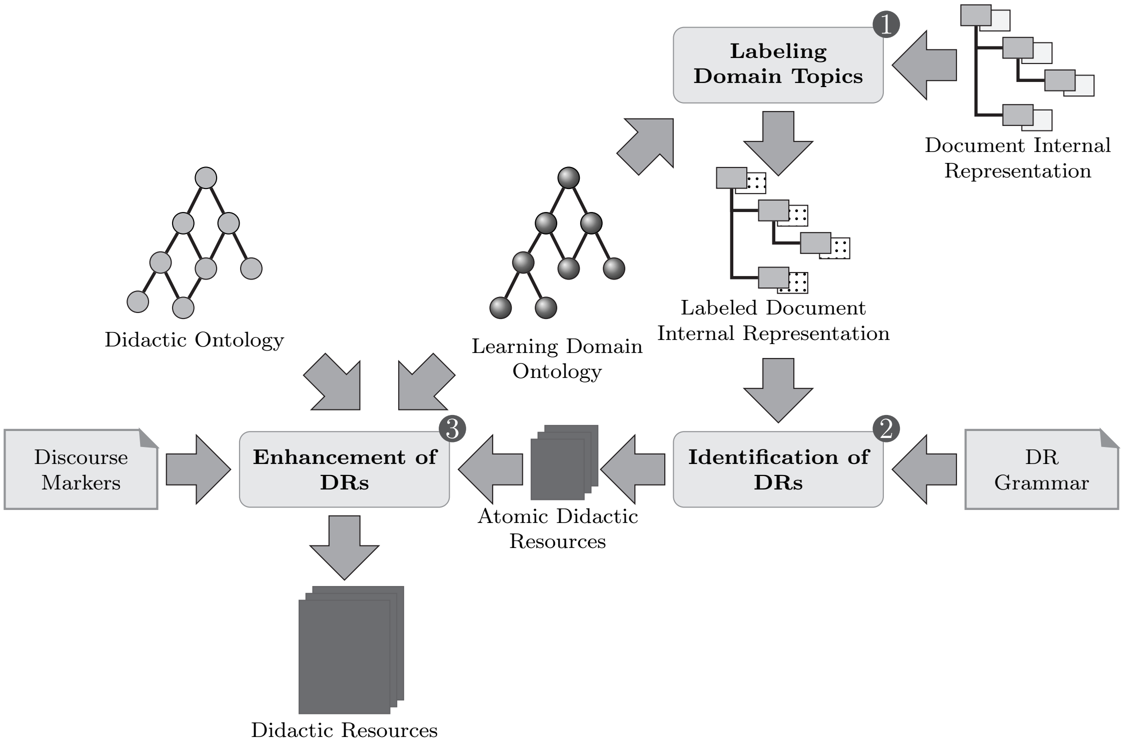


Figure 6.2 – Generation of Didactic Resources (from Larrañaga, 2012)

Given that the DRs identified by the DR Grammar are usually quite simple, they are enhanced in two ways to make them more accurate. On the one hand, consecutive DRs are combined if they are similar. On the other hand, and to keep the cohesion of the DRs, previous fragments are added to each DR if it contains references to those previous DRs or sentences. The cohesion maintenance relies on the use of Discourse Markers, i.e., words or expressions that connect part of a text with its context.

6.1.2 From Didactic Resources to Learning Objects

The possibility of retrieving the desired LO from a large set or a LOR is a key issue to promote the use and reuse of LOs. The selection of a suitable LO is highly influenced by how appropriate the metadata that describes it is (Cardinaels, 2007). While the manual creation of metadata can be considered for annotation of a single LO, it is not an option for larger deployments, where a considerable number of LOs have to be managed (Cardinaels et al., 2005). Furthermore, semiautomatic metadata generation by using ontologies can overcome metadata inconsistency problems (Kabel et al., 2004).

The LO metadata for each LO is generated automatically, first, using Samgi (Meire et al., 2007), an automatic metadata generator. Then, the metadata is enhanced with more information that has been extracted during the DR generation to improve some metadata elements (*Keywords* or *Learning Resource Type*). Most keyword annotation applications use statistical methods and rely on the frequency of the terms in the analyzed text, but do not consider semantic relationships among the topics. Thus, an ontology-based metadata enhancement process is carried out. The LDO and the identified domain topics in the LO are used to get a more accurate *Keyword* list, as the semantics of the relationships are taken into account.

The *Learning Resource Type* is specified in terms of the ALOCOM ontology (Verbert et al., 2005), which represents a content model for the LOs and its components. The generated LOs can be *definitions*, *examples*, *principle statements*, *problem statements*, *theories*, and *facts*, which are kinds of LOs described in the ALOCOM ontology. To determine the *Learning Resource Type*, the rules of the DR Grammar met by the content of the DR are used. As these rules may identify different kinds of DRs, the *accuracy* of the rules (percentage of times that the rule correctly identifies a DR) is used to determine which the most plausible kind is, which is therefore selected as the *Learning Resource Type* for the annotated LO. Figure 6.3 shows an example in which several DR rules, each recognising a different kind of LO, triggered. In the *Learning Resource Type* identification process, as the DR rule with highest confidence is the one which recognised a *definition*, this kind of LO was determined.

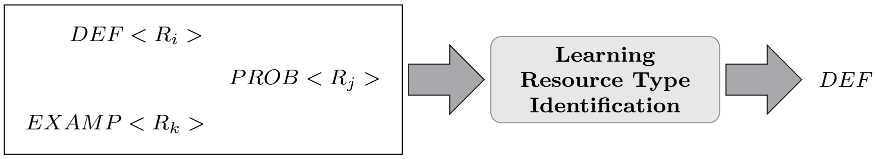


Figure 6.3 – Learning Resource Type Identification

6.1.3 Learning Objects Storage

Once the LOs and their preview files have been generated, they are inserted in the LOR to allow their retrieval and use in TSLSs. The LO publishing service is based on the Simple Publishing Interface (SPI) specification (Ternier et al., 2008). The LOR can be queried to find the appropriate LOs using Simple Query Interface (SQI) (Simon et al., 2005). When the LO is composed, all its components are also appropriately labeled and stored in the LOR, as they might be useful in certain contexts.

6.2 Elicitation of Learning Objects for the English Language

The *ErauzOnt* module in *DOM-Sortze* is responsible for the automatic extraction of LOs from electronic textbooks following the approach described above. The framework aims to be applicable on any document no matter which domain it relates to. None of its components relies on implicit domain-specific knowledge. All the used domain-specific knowledge are the domain topics and the relationships among those topics described on the LDO. Although *ErauzOnt* was originally evaluated with textbooks written in the Basque language, it was designed to be easily extended to support new languages. In fact, the work here presented required the enhancement of *ErauzOnt* to support the English language for eliciting LOs (Conde et al., 2012).

Table 6.1 – Example of a Pattern that Allows Identifying Definitions

	Basque	**English**
Pattern	@Topic *definition* (DAT[a]) **deitu**	*Definition* **To Be called** @Topic
Example	Unibertsoa *astro guztien multzoari eta betetzen duten espazio***ari deitzen zaio**.	*The whole set of celestial bodies and the space they fill* **is called** Universe.

[a] The (DAT) states that the text of the description is in dative case, which is expressed with the **-ari** suffix in the example

ErauzOnt relies on NLP techniques to identify the relevant DRs in the textbooks, so an analyser must be integrated for each supported language. *ErauzOnt* has been enhanced to use FreeLing (Padró and Stanilovsky, 2012), an analyser that supports several languages such as English or Spanish. It was also necessary to define the DR Grammar, which contains the syntactic patterns used in English for the DRs, and the Discourse Markers for English. Table 6.1 and Table 6.2 show examples of rules that allow the identification of *definitions* and *examples* respectively adapted from Basque to English. A total of 46 rules have been adapted from Basque to English (21 rules for *definitions*, 8 for *examples*, 2 for *facts*, 10 for *principle statements*, 1 for *theories*, and 4 for *problem statements*). In addition, 18 Discourse Markers have been identified for English. *ErauzOnt* uses the appropriate resources, i.e., NLP analyser, DR Grammar and Discourse Markers, for each document according to the language it is written in.

Table 6.2 – Example of a Pattern that Allows Identifying Examples

	Basque	**English**
Pattern	**adibidez,** @Topic	**for instance,** @Topic
Example	Uretan, **adibidez,** hidrogeno eta oxigeno atomoak daude.	**For instance,** there are hydrogen and oxygen atoms in water.

Next, the experiment conducted to validate the proposal is presented.

6.2.1 Experiment

Once *ErauzOnt* was enhanced to support English, it was tested to assess its performance. The *Principles of Object-Oriented Programming* textbook (Wong and Nguyen, 2010), which tackles the basic concepts of Object-Oriented Programming, was used for this evaluation. The main goal of the experiment was the evaluation of the acquisition of text-based LOs, so an adapted version of the textbook, in which the images were removed, was processed instead of the original document. The analysed book consists of 67 pages and 29,300 words.

For this experiment, the teachers of the subject were requested to define the LDO that describes the topics to be learnt, as well as the pedagogical relationships among the topics. This LDO was then used by *ErauzOnt* to extract the LOs and store them in the LOR. The extracted LOs were evaluated using both the *Gold-standard* and expert validation approaches. To build the *Gold-standard*, the teachers of the subject manually analyzed the textbook to identify and label the set of DRs (*definitions*, *examples*, etc.) that they would like to use for mastering the main topics of the subject. A group of instructional designers evaluated the automatically extracted LOs considering both the *Gold-standard* and the appropriateness of the extracted LOs for their use in education contexts.

Given that this was the first experiment with *ErauzOnt* over documents written in English, the performance of DR Grammar was also tested. Before the experiment, the DR Grammar was run first on a sample of text fragments to empirically determine the confidence of the DR Rules in this evaluation experiment.

6.2.1.1 Evaluation of the Didactic Rresource Grammar for English

The DR Grammar was evaluated by analyzing the gathered atomic LOs, i.e., the finest-grained LOs. Each LO was inspected to measure the accuracy of the DR Grammar by identifying which rules were used and whether they worked properly or not.

Table 6.3 shows the statistics about the evaluation of the DR Grammar. The DR Grammar is able to identify *definitions*, *examples*, *problem statements*, *principle statements*, *facts* and *theories*. However, not every kind of DR is always used. Neither *facts* nor *theories* were found in the analysed textbook. The DR Grammar built for identifying the syntactic patterns commonly used in DRs achieved 80.09% *accuracy*. The average of the rules ranges from 100.00% for the *examples* to 58.33% for the *problem statements*. The DR grammar achieved similar results to the previously conducted experiments over textbooks in Basque (Larrañaga et al., 2012), except that the *accuracy* for *problem statements* was considerably lower, mainly because imperative cases, frequently used to state *problem statements*, are easier to identify in Basque, which uses an auxiliary verb for that purpose. The identification of the *problem statements* in English mainly relies on the appearance of keywords such as *exercise*.

	Definitions	**Examples**	**Prob. Stat.**	**Princ. Stat.**	**Total**
Found	164	1	12	49	226
Correct	138	1	7	35	181
Accur. (%)	84.15	100.00	58.33	71.43	80.09

Table 6.3 – Accuracy of the Didactic Resource Grammar

6.2.1.2 Evaluation of the Learning Object Acquisition Process

In the next step, the gathered LOs were evaluated. The evaluation was carried out following a *Gold-standard* approach, i.e., by comparing the manually identified DRs with those automatically gathered by *ErauzOnt*. The evaluation of the gathered LOs considered both their appropriateness (*precision*) and the quantity of the manually defined DRs that were automatically identified (*recall*).

In order to obtain the *recall* of the LO acquisition process, the automatically gathered LOs were compared to the manually identified ones. Table 6.4 summarises the results. The teachers identified 54 DRs, 35 *definitions*, 2 *problem statements* and 17 *combined DRs*, i.e., DRs that entail two or more DRs of different kinds. *ErauzOnt*

	Definitions	**Problem Stat.**	**Combined DRs**	**Total**
Real	35	2	17	54
Found	22	2	17	41
Recall (%)	62.86	100.00	100.00	75.93

Table 6.4 – Recall of the Learning Object Acquisition Process

achieved a 75.93% *recall*, i.e., 41 of 54 manually identified DRs were automatically gathered. 62.86% of the *definitions*, 100.00% of the *problem statements*, and 100% of the *combined DRs* were automatically gathered. *Problem statements* proved to be easy to find, while *definitions* were more difficult. *Problem statements* are presented using verbs in the imperative case or keywords such as exercise, while *definitions* may appear in many different forms that make them more difficult to find.

Determining the *precision* was not so straightforward, and all the gathered LOs and their components had to be analysed. While a particular LO could be the most appropriate one for a certain context, one of its component LOs or a more complex LO, a composite LO that comprises it, might fit better in other situations. Therefore, each generated LO was evaluated following an *expert validation* approach by determining whether it was valid, not only considering the subject for whom the textbook was analysed but any other context.

	Definitions	**Problem Stat.**	**Combined LOs**	**Total**
Found	140	2	229	371
Correct	121	2	199	322
Precision (%)	86.43	100.00	86.90	86.79

Table 6.5 – Precision of the Learning Object Acquisition Process

Table 6.5 summarises the information extracted from the analysis of the automatically elicited LOs. *ErauzOnt* gathered 371 LOs, 140 *definitions*, 2 *problem statements*, and 229 *combined LOs*, i.e., LOs that comprise LOs of different kinds.

As mentioned above, more than one rule can trigger in the same LO, even if they identify different kinds of LOs. In this experiment, although the DR Rules for the identification of *principle statements* fired, the corresponding text fragments were part of other kinds of LOs. Therefore, no LOs of this kind were identified. Furthermore, the DR Rules for the elicitation of *problem-statements* triggered with a relatively low accuracy. However, most of those activations co-occurred with other rule fires, and the grammar performed very well in terms of both *precision* and *recall* regarding this kind of LOs.

The overall *precision* achieved was 86.79%, i.e., 322 of the 371 LOs were considered usable for this course or any course that might be developed in the future. *Problem statements* obtained 100% precision, while *definitions* got 86.43% and *combined LOs* 86.90%. Considering these results, the pattern-based approached used by *ErauzOnt* to gather LOs from electronic textbooks proved to be accurate and useful.

6.3 Wikipedia and WordNet as Sources of Educational Contents

In Section 3.4.1, some characteristics of Wikipedia have been described. This time, some features that make Wikipedia a particularly interesting resource to fulfil the objectives pursued in this chapter are described. Wikipedia can be considered as the largest and oldest multilingual and participatory encyclopaedia in the Internet. In a few years, Wikipedia has become an unprecedented phenomenon of collaborative construction of knowledge. It is the world's largest encyclopedia and grows daily thanks to the contributions of millions of collaborators who selflessly share their knowledge on various themes in different languages. Wikipedia can be used to define an interlingual or universal concept space (Sorg and Cimiano, 2012). But this constructive mode raises doubts about the veracity of their contents, control mechanisms and veracity of contributions and, ultimately, the validity of Wikipedia as a source and resource.

Just a few years ago, Wikipedia was seen as a barbarian invading the ivory tower; now, an increasing number of academics recognize that it can be used as an effective teaching tool (Konieczny, 2012). Although some doubts about the validity of Wikipedia as a source exists, some of the academics are incorporating Wikipedia assignments into their classes, pursuing learning outcomes that involve at least one of the following areas: writing skills, information literacy, research skills, critical thinking, collaboration, translation skills, or literature review skills (Dunican, 2013). Lim (2009) pointed out that educators and librarians need to provide better guidelines for using Wikipedia, rather than prohibiting Wikipedia use altogether.

The thesis here presented does not want to analyse the validity of Wikipedia contents as good or bad educational resources. In no case will the validity of the extracted content be analysed from an educational point of view. This work only considers Wikipedia as a large-coverage multilingual resource that shows a high potential for being successfully employed in the task of retrieving multilingual LOs.

Wikipedia is considered as a particularly interesting resource due to the fact that (1) it contains a broad coverage of topics from multiple fields or domains, and (2) it contains knowledge about those concepts in different languages. Wikipedia is segregated by languages. Cross-language links of Wikipedia are one of the main features exploited in our proposal. Cross-language links connect equivalent articles or categories across languages, so, concepts in Wikipedia that are aligned across languages will provide us the necessary target articles or documents to process. However, not all content is linked across languages and, therefore, it is neither possible to identify LOs in all the considered languages nor the same set of LOs in all the considered languages. It is known that some topics are described in some languages and not in others, or some topics are described better that others, furthermore, the same topic is described better in one language than in another. Taking those facts into account, it will be possible, for example, to extract a definition for the topic X in English and not in Basque, because the cross-link for topic X between English and Basque does not exist. In addition, although the cross link between both languages exists, it could happen that the article in English is more complete and includes LO types not considered in the Basque language, or just the opposite. Moreover, in the same language, and depending on the completeness of the article, it would be possible to identify *definition* and *bibliographical reference* LO types for the topic Y and only *definition* for topic Z.

Every Wikipedia article entails a brief *definition* or description about the topic the article relates to, deeper information about the topic. Most articles also include some *References*, *Bibliography*, *See also* or even *Further readings* blocks that might link to more interesting information on the topics.

Although, to a lesser extend, in addition to Wikipedia, WordNet can also be used as a source for gathering educational resources. WordNet is a lexical database with manually defined semantic relationships and *definitions*. Therefore, it can be considered as a source to elicit new *definitions*.

Specifically, in this chapter Wikipedia and WordNet are used to:

- Enrich the set of already identified types of LOs in the English language, such as *definitions*. For each article/concept, Wikipedia and WordNet provide a manually written *definition* that can be extracted to enrich the set of automatically built LOs.

- Provide LOs in different languages. As Wikipedia articles provide mappings to articles referring to the same concept in other languages, it could be used in order to provide multilingual support.

- Identify new types of LOs such as *bibliographical references* using information provided in Wikipedia articles.

6.4 LiLoWi: a Multilingual Learning Object Extractor

Throughout this section, *LiLoWi* is described. This is an extractor of LOs that allows the elicitation of new LOs, including some multilingual ones, for the topics of the LDO. *LiLoWi* uses both WordNet and Wikipedia for the extraction of new LOs. As pointed out above, in WordNet, each concept has an associated *definition*, along with the semantic relationships with other concepts. Wikipedia, on the other hand, provides a wider bunch of information about a concept in its corresponding article: *definition*, *historic information*, *bibliographic references*, etc. The content of the Wikipedia articles is organised in so-called blocks. Furthermore, the articles follow a predictable layout which states that the *definition* of the topic referred by the article is in the first block. Obviously, the blocks vary from one article to another, but many of them are quite common and can be used for the elicitation of new kinds of LOs. A preliminary analysis was carried out on a small sample of Wikipedia articles to determine which blocks might be useful for such a purpose. The *references* block is used to present resources that sustain any claim that eventually might appear in the article. In many cases, they relate to the concept described in the article indirectly. However, *further reading* or *bibliography* do recommend additional readings or sources of information closely related to the concept being described. Therefore, those blocks, along with the *definition* seem to be appropriate for the elicitation of LOs from Wikipedia.

In order to carry out the LO elicitation from Wikipedia and WordNet, the topics have first to be mapped to their corresponding entries in those resources. In *LiDom Builder*, the topics have already been mapped to Wikipedia (see Section 4.1.2.2) and WordNet (see Section 5.2.1).

To extract the new LOs for the topics of the LDO, *LiLoWi* retrieves the information from WordNet and Wikipedia through their corresponding LO Extractors –*WordNetLOExtractor* and *WikipediaLOExtractor*. The *WordNetLOExtractor* uses the MIT Java Wordnet Interface (JWI[1]) to extract the definitions of the topics from WordNet, whilst *WikipediaLOExtractor* uses Wikiminer (Milne and Witten, 2013) to gather the information from Wikipedia. To this end, *WikipediaLOExtractor* carries out the following process for each topic (see Figure 6.4):

[1] http://projects.csail.mit.edu/jwi/

1. Filter the blocks that are suitable for LO elicitation.

2. Extract the *definition* for the topic. Given the multilingual nature of Wikipedia and the predictable layout of the articles, where the *definition* is placed first, the translations of the *definitions* are also extracted building multilingual *definitions*.

3. Additional LOs, *references*, are built from the other blocks.

Finally, every generated LO is automatically annotated with metadata and stored in the LOR to facilitate their further retrieval and the procedure described in Section 6.1.2 and Section 6.1.3 is used.

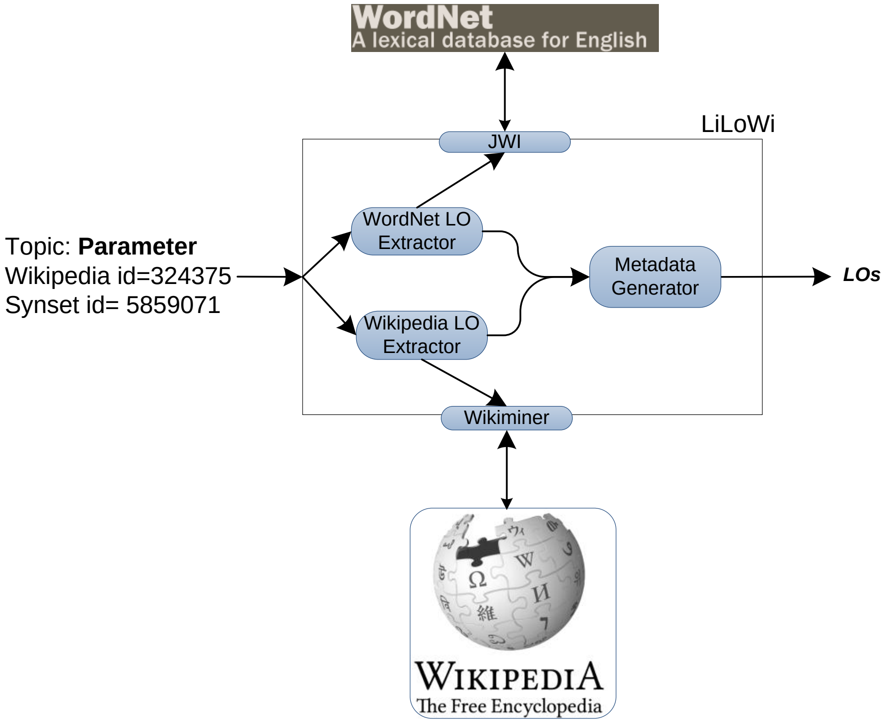


Figure 6.4 – Wikipedia/WordNet-based Learning Object Generation

6.5 Evaluation

In this section, the experiment conducted to evaluate *LiLoWi* is depicted. To test its performance, *LiLoWi* was run on the *Principles of Object-Oriented Programming* textbook (Wong and Nguyen, 2010) using the same LDO developed for the experiment described in Section 6.2.1. Evaluating the quality of the content in Wikipedia and WordNet is beyond the scope of this experiment. Furthermore, the validity of those resources for education has already been highlighted in Section 6.3. Therefore, the experiment here conducted consisted of measuring how much the use of *LiLoWi* enhanced the LO coverage for the LDO topics, and how many multilingual LOs were elicited.

In the experiment presented in Section 6.2.1, *ErauzOnt* gathered 371 LOs (322 of them valid) that cover 21 of the 82 LDO topics (25.61%). Regarding the elicitation of *definitions*, it was able to extract *definitions* for 19 topics (19.51%). This information is summarised in Table 6.6.

	Total	**Definitions**
Number of topics	21	19
Topic coverage(%)	25.61	19.51

Table 6.6 – Learning Object-Topic Coverage Using ErauzOnt

This time, using Wikipedia and WordNet as additional resources, *LiLoWi* was able to extract *definitions* for 46 topics (56.10%). Whilst Wikipedia provided *definitions* for those 46 topics, WordNet provided *definitions* for 12 of them. 34 of the topics *LiLoWi* provided with *definitions* (41.46%) referred to topics that *ErauzOnt* was not able to provide with a *definition*.

Regarding the provision of multilingual *definitions* (see Table 6.7), *LiLoWi* extracted *definitions* in Spanish for 36 topics (43.90%), 9 in Basque (10.97%), and 36 in French (43.90%).

Finally, regarding the acquisition of additional LOs, i.e., *references*, *LiLoWi* elicited this kind of LOs for 12 topics (14.63%).

All in all, *LiLoWi* was able to provide 29 topics (35.37%) with new LOs that had not been provided with LOs by *ErauzOnt*.

	Definitions				**Refs.**
	English	**Spanish**	**Basque**	**French**	
Number of topics	46	36	9	36	12
Topic coverage(%)	56.10	43.90	10.97	43.90	14.63

Table 6.7 – Learning Object-Topic Coverage Using LiLoWi

In summary, it can be concluded that the integration of Wikipedia and WordNet remarkably improves the elicitation of LOs, as regards the topic coverage. The benefit of using WordNet in the experiment was restricted by the proportion of topics that could be mapped to that knowledge base. Regarding the generation of multilingual LOs, *LiLoWi* provides a first approach towards this final aim. As can be seen in Table 6.7, multilingual *definitions* could be extracted for a remarkable proportion of topics. The coverage of multilingual resources achieved is correlated with the spread and presence of languages in Wikipedia.

6.6 Comparison with Other Approaches

As presented in this chapter, and with regard to the identification of LOs from textbook bodies written in English, *LiDom Builder* follows the same approach used by *DOM-Sortze* (Larrañaga, 2012; Larrañaga et al., 2014) to identify the following kinds of LOs: *definition*, *example*, *fact*, *theory*, *principle statement*, and *problem statement*. To deal with the English language, *LiDom Builder* incorporates the FreeLing analyser (Padró and Stanilovsky, 2012) and both a DR Grammar for English and the Discourse Markers for English.

To determine the *Learning Resource Type*, the rules of the DR Grammar met by the content of the DR are used. As these rules may identify different kinds of DRs, the *accuracy* of the rules (percentage of times the rule correctly identifies a DR) is used to determine which the most plausible kind of resource type is. The work of Jain and Pareek (2012) is the most similar to the approach followed in *LiDom Builder* to identify *Learning Resource Types*. They also use a pattern-based method to identify the type of resource from sentences written in English. Trigger patterns exist for identifying LO types corresponding to *explanation*, *application*, *experiment*, *exercises*, and *case study*. One by one and for each sentence, the presence of a pattern pre-stored in the pattern-base is checked. Separate counters keep track of the number of patterns identified under each category. Counter value under the corresponding category is incremented if the pattern listed under that category is found in the sentence.

Roy et al. (2008) used a Machine Learning approach to classify documents as learning materials. Identifying some surface level features of the text, such as the occurrence of a set of specific verbs, trigger words, phrases, and special characters, they classify learning materials into three categories using a neural network: *narrative text*, *experiment*, and *exercise* (a subset of the *Learning Resource Types* identified in IEEE LOM (LTSC, 2001) 5.2 specification).

With regard to particular *Learning Resource Types*, only *definition* identifications have been found in the literature. However, although *definition* extraction is an important task in the NLP area, most works are oriented towards the automatic creation of glossaries and dictionary building, and only a few works dealt with the extraction of *definitions* as a way to get educational resources. Westerhout (2009) uses a sequential combination of a rule-based approach and Machine Learning to extract *definitions* in Dutch that will constitute glossaries within e-learning. As a first step a grammar is used to match sentences with a *definition* pattern and thereafter, Machine Learning techniques are applied to filter out those sentences that, despite having a *definition* pattern, do not qualify as *definitions.* They divide *definitions* into four categories: *is-definitions*, *verb-definitions*, *punctuation definitions*, and *pronoun definitions*. *Is-definitions* are the definitions in which a form of the verb '*To Be*' is used as a connector. In *verb-definitions*, a verb other than '*To Be*' is used as connector. In *punctuation definitions*, a punctuation character is used as connector, and, finally, in *pronoun definitions* relative or demostrative pronouns are used to point back to a defined term that is mentioned in a preceding sentence.

Finally, it must be pointed out that, to the extent of our knowledge, from knowledge bases such as Wikipedia or WordNet, no work allows the automatic extraction of educational contents, appart from *definitions* for linguistic purposes.

6.7 Conclusions

In this chapter, the elicitation of multilingual LOs has been described. The chapter first focused on the language independence of the proposal of *DOM-Sortze* regarding the acquisition of LOs. To this end, the DR grammar used for the identification of meaningful text fragments with educational purposes was adapted to English, and the Discourse Markers for English were identified. An experiment was then conducted to verify the appropriateness of the approach.

In addition, this chapter also presented *LiLoWi*, a module for the extraction of Multilingual LOs that uses both Wikipedia and WordNet as additional resources. *LiLoWi* extracts additional LOs from those resources, while taking advantage of the multilingual nature of Wikipedia to enhance the LO acquisition process with the provision of multilingual *definitions.* Once more, an experiment was carried out to measure how *LiLoWi* enhanced the topic coverage with new LOs.

The next chapter presents the final remarks of this dissertation along with the future work lines identified.

7 Conclusions and Future Work

This final chapter summarises the main contributions and future research lines resulting from the current study, and relates the contributions to the main research questions addressed at the beginning of the thesis in Chapter 2.

7.1 Results and Contributions

Throughout this dissertation, the analysis, design and evaluation of *LiDom Builder*, a framework for the automatic development of *Multilingual Domain Modules* from electronic documents for TSLSs, have been described. *LiDom Builder* employs NLP and Machine Learning techniques, together with multilingual resources such as Wikipedia or WordNet, for the knowledge acquisition processes.

LiDom Builder can be considered an evolution of *DOM-Sortze* (Larrañaga, 2012; Larrañaga et al., 2014) in the transition from *Monolingual Domain Modules* towards *Multilingual Domain Modules*. To this end, *LiDom Builder* incorporates a mechanism capable of representing the domain from a multilingual point of view. A *Multilingual Domain Module* entails a LDO with topics labelled in diverse languages, pedagogical relationships among those topics, and LOs in those languages. *LiDom Builder* provides the way to represent the topics of the domain in every supported language. Each topic is linked to its equivalent label in the corresponding language. In addition, the formalism used in *LiDom Builder* enriches the metadata of each LO to describe the link to their equivalents in the other languages.

In *LiDom Builder*, the *Domain Module* is initially gathered and built from a document written in a specific language, and multilingual resources are used to obtain the correspondent topics and LOs in other languages. In the presented work, textbooks in the English language have been used as main sources of information for tuning and evaluation purposes: *Principles of Object Oriented Programming* textbook (Wong and Nguyen, 2010), *Introduction to Astronomy textbook* (Morison, 2008), and *Introduction to Molecular Biology* textbook (Raineri, 2010). With regard to the multilingual knowledge bases, Wikipedia, WordNet and other resources derived from Wikipedia have been used. There are three main modules of *LiDom Builder* that contribute to the *Multilingual Domain Module* acquisition process from textbooks: *LiTeWi* and *LiReWi* are needed to build the multilingual LDO, whilst *LiLoWi* allows the elicitation of multilingual LOs.

Next, the main characteristics of each module are briefly pointed out.

1. *LiTeWi* (Conde et al., 2015) is the module responsible for the elicitation of multilingual terms for Educational Ontologies from electronic documents. It combines different approaches such as TF-IDF, KP-Miner, CValue and Shallow Parsing Grammar for the unsupervised term extraction using Wikipedia as a knowledge base. The approach carried out by *LiTeWi* entails three main steps: the identification of the topic candidates; the combination and the refinement of the results to obtain the set of terms; and, finally, the mapping of the terms to other languages in Wikipedia.

2. *LiReWi* (Conde et al., submitted) is the module that implements a method for the elicitation of pedagogical relationships for Educational Ontologies from electronic document bodies. It combines shadow parsing techniques in addition to several knowledge bases such as Wikipedia, WordNet, WikiTaxonomy, WibiTaxonomy and WikiRelations to elecit *isA*, *partOf*, *prerequisite* and *pedagogicallyClose* relationships. *LiReWi* also performs a three-step procedure to fulfil its task: first, all the topics are mapped to the diverse knowledge bases that will be used to identify the relationships; then, several relationship extractors, each using a different approach, are concurrently run to elicit candidate relationships; and, finally, the results are combined and filtered to obtain the final set of pedagogical relationships.

   In *LiDom Builder* the process of eliciting structural relationships (*isA*, *partOf*) from document outlines has also been enhanced, with the inclusion of Wikipedia as an additional resource (Conde et al., 2014).

3. *LiLoWi* is the module that enables the elicitation of new LOs, including some multilingual LOs, from both the original textbook body and different knowledge bases such as Wikipedia or WordNet. Once each topic of the LDO is mapped

to Wikipedia and WordNet, LiLoWi retrieves the information from those two resources using their corresponding LO Extractors.

Before incorporating Wikipedia and WordNet to the LO acquisition process, the validity of the proposal presented in *LiDom Builder* to incorporate the English language has also been considered and tested (Conde et al., 2012).

Regarding evaluation issues, it is worth mentioning that *LiDom Builder* modules were individually tested and validated using both a *Gold-standard* approach and an expert validation. In addition, how integrating WordNet and Wikipedia in the LO elicitation improved LO elicitation regarding the topic coverage has also been measured. In all the cases the obtained results are very satisfactory.

To conclude, there are four main contributions of *LiDom Builder* to the *Multilingual Domain Module* acquisition area:

- Provision of a suitable mechanism to represent *Multilingual Domain Modules*.
- Development of *LiTeWi*, a module for the elicitation of multilingual terminology for Educational Ontologies. The version for Spanish and English is available at `https://github.com/Neuw84/LiTe`.
- Development of *LiReWi*, a module for the elicitation of pedagogical relationships for Educational Ontologies. For this module, a Wikipedia/WordNet mapper is available at `https://github.com/Neuw84/Wikipedia2WordNet`.
- Development of *LiLoWi*, a module for the elicitation of multilingual LOs.

7.2 Future Research Lines

This sections describes the future research lines opened by this thesis. They range from improving the modules in *LiDom Builder* to the automatisation of the integration of new languages, or the inclusion of new types of LOs in the LO generation.

7.2.1 Improvements in LiDom Builder Modules

Three are the main modules involved in *LiDom Builder*: *LiTeWi*, *LiReWi* and *LiLoWi*.

To improve the elicitation on multilingual terminology in *LiTeWi*, additional techniques such as Likey (Paukkeri et al., 2008), RAKE (Rose et al., 2010) or DP-SEG (Koilada et al., 2012), could be integrated to enhance the candidate extraction process. In addition, the disambiguation process could be improved by using, for example (Pohl, 2012), to determine the most appropriate or promising features to train the Wikiminer relatedness classifier. Finally, the filtering process could be enhanced using more sofisticated techniques such as voting or supervised Machine Learning algorithms. Clustering techniques such as spectral clustering might also be useful to filter non-related words (Aggarwal and Zhai, 2012).

As regards *LiReWi*, the elicitation of the relationships could enhance integrating multilingual knowledge bases such as MENTA (Melo and Weikum, 2014), BabelNet (Navigli and Ponzetto, 2012), Yago3 (Mahdisoltani et al., 2014), or Multilingual WordNet (Bond and Foster, 2013). *LiReWi* currently extracts relationships from the document body or from additional resources such as Wornet or Wikipedia-based resources. Wikipedia-based resources provide information considering the information contained in the English Wikipedia. The aforementioned multilingual resources provide further information. For instance, MENTA includes information from particular Wikipedia languages. In this way, the mapping to the English Wikipedia articles would not be a requirement anymore.

Finally, to improve the LO elicitation, the use of *ErauzOnt* could be extended to additional documents, for instance to Wikipedia articles.

7.2.2 Automatising the Inclusion of New Languages

Although the modular design of *LiDom Builder* facilitates the inclusion of a new language, some resources must be defined, in particular the heuristics and the grammars that allow the knowledge elicitation and the Discourse Markers for that language.

Automatising the development of such kinds of resources will remarkably reduce the workload in the integration of a new language. In the last few years, great advances have been made in Machine Translation. The research in that field might help to semi-automatically develop the grammars and heuristics for a new language from those already defined for a particular language.

Furthermore, similar structures or equivalent patterns have been observed in the supported languages. Therefore, a meta model describing the generic patterns could be defined and rule-based transformations applied to obtain the specific grammars and heuristics for a particular language.

7.2.3 Using New Approaches for the Generation of Multilingual Learning Objects from Monolingual Learning Objects

The current version of *LiDom Builder* facilitates the acquisition of multilingual resources such as Wikipedia. Furthermore, given the multilingual nature and the layout of Wikipedia, *LiDom Builder* is able to generate multilingual *definitions* from Wikipedia. Using *ErauzOnt* on Wikipedia, or other additional resources, would allow the identification of additional monolingual LOs. To generate multilingual LOs from these resources, two different approaches could be applied.

LiDom Builder could try to identify LOs that are equivalents or translations in other languages. To this end, different means will be explored. For example,

- Latent Semantic Analysis (LSA) would be used to generate a model of each LO, and this model would be translated using Machine Translation techniques to obtain its equivalents in other languages. If a similar model were found for the translated model, then the equivalence between their corresponding LOs would be inferred.
- Additionally, another Machine Translation based approach might be also explored. To determine if two LOs, say LO_1 in English and LO_2 in French, are equivalent, *LiDom Builder* could take advantage of Machine Translation techniques by generating their automatic translations before comparing them. If the translated LO_1 (LO_{t1}) were similar to LO_2, or the translation of LO_2 (LO_{t2}) were similar to LO_1, they could be considered equivalent. Diverse similarity and text reuse metrics would be tested in this approach.

7.2.4 Concept Map-Based Learning Object Generation

A concept map is a diagram showing the relationships among concepts. Concept maps are graphical tools for organising and representing knowledge (Novak and Cañas, 2008). They include nodes (concepts), usually enclosed in circles or boxes, and relationships between concepts, connection lines linking at least two concepts that establish propositions. Both nodes and links can be labelled with a key or brief text that adds semantics to them. Concept maps have their origin in the learning movement called constructivism. The concept mapping technique was developed by Novak and his research team at Cornell University in the 1970s (Novak, 1977). The fundamentals of concept mapping are in Ausubel's learning and assimilation theories. The former is based on the assumption that meaningful learning occurs when the new concepts are linked to familiar concepts existing in the learner's cognitive structure (Ausubel, 1968). The latter is sustained in the hierarchical structure of a concept map from an abstract level to a more specific level helping in the comprehensiveness of the information (Ausubel et al., 1978). Until now, in the educational area, concept maps have been mainly used to support learning.

The GaLan research group has already some experience using concept maps in the learning process. From CM-ED (Concept Map Editor) to Elkar-CM (Collaborative Concept Map Editor), the group has tried using concept maps both individually and collaboratively in monolingual and multilingual educational contexts (Arruarte et al., 2012; Calvo et al., 2013; Elorriaga et al., 2013). Both CM-ED and Elkar-CM allow defining multilingual concept maps through a localised view mechanism.

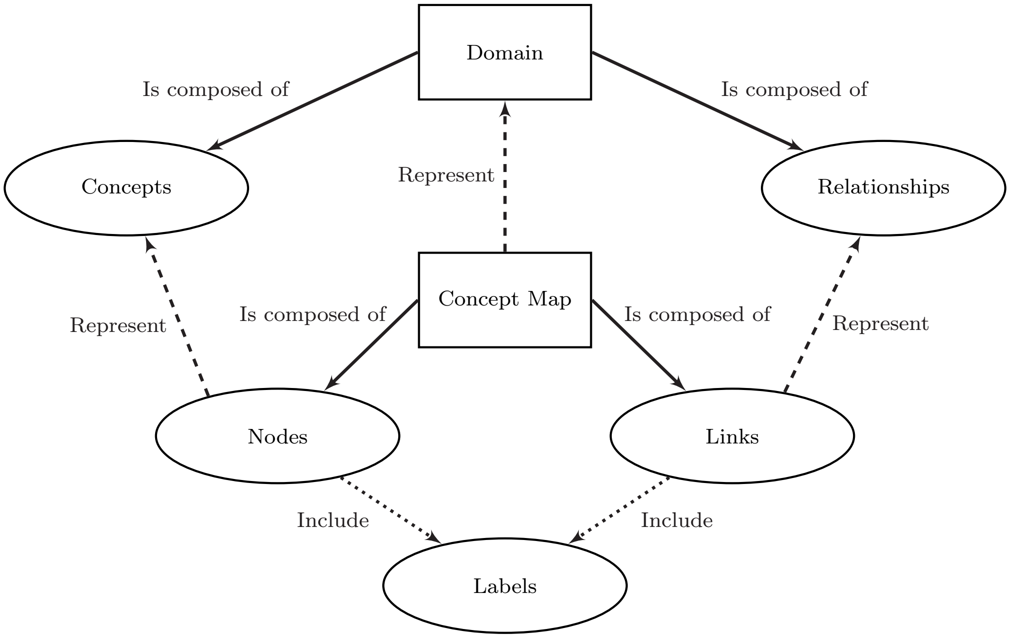


Figure 7.1 – Concept Map Example

Figure 7.1 shows an example of a concept map. It can be observed that its representation is not far from the representation used in *LiDom Builder* to visualize the LDO, i.e, the Learning Domain Ontology. The elicitation of new types of relationships in *LiDom Builder*, relationships different from the currently identified pedagogical relationships, would allow the automatic generation of concept maps related to the domain considered in the textbook that *LiDom Builder* used as a source. The concept maps, along with their localised views, would constitute a new kind of multilingual LOs.

7.2.5 Supervision of the Multilingual Domain Module Generation

The construction of *Domain Modules*, either monolingual or multilingual, requires the supervision of the instructors that will use them in their learning sessions. On the one hand, automatic information elicitation might be error-prone and, therefore, they must correct any mistake in the automatic process. On the other hand, the instructors might want to adapt the automatically generated *Domain Modules* to their preferences. In *DOM-Sortze*, a collaborative tool was included to support this supervision step, *Elkar-DOM*. Furthermore, this tool was responsible for adjusting the confidence levels of the heuristics and rules according to the corrections made by the instructors. However, *Elkar-DOM* was not prepared for the supervision of *Multilingual Domain Modules*, and, thus, it needs to be enhanced to deal with such kinds of *Domain Modules*.

Bibliography

Aduriz, I., Aldezabal, I., Alegria, I., Artola, X., Ezeiza, N., and Urizar, R. (1996). "EUSLEM: A Lemmatiser / Tagger for Basque." In: *Proceedings of the 7th EURALEX International Congress on Lexicography, EURALEX 1996*. Vol. 1. Göteborg, Sweden, pp. 17–26.

Aggarwal, C. and Zhai, C. (2012). "A Survey of Text Clustering Algorithms." English. In: *Mining Text Data*. Ed. by C. C. Aggarwal and C. Zhai. Springer US, pp. 77–128. ISBN: 978-1-4614-3222-7. DOI: 10.1007/978-1-4614-3223-4_4. URL: http://dx.doi.org/10.1007/978-1-4614-3223-4_4.

Agirre, E. and Soroa, A. (2009). "Personalizing PageRank for Word Sense Disambiguation." In: *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2009*. Athens, Greece: The Association for Computer Linguistics, pp. 33–41.

Aherne, A. and Vogel, C. (2003). "Wordnet enhanced automatic crossword generation." In: *Proceedings of the 3rd International Wordnet Conference*, pp. 139–145.

Aldabe, I. (2011). "Automatic Exercise Generation Based on Corpora and Natural Lenguaje Processing Techniques." Doctoral dissertation. University of the Basque Country (UPV/EHU).

Aldabe, I. and Maritxalar, M. (2014). "Semantic Similarity Measures for the Generation of Science Tests in Basque." In: *IEEE Transactions on Learning Technologies* 6, pp. 1–14. ISSN: 1939-1382. DOI: 10.1109/TLT.2014.2355831. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=%5C&arnumber=6894225.

Alegria, I., Gurrutxaga, A., Lizaso, P., Saralegi, X., Ugartetxea, S., and Urizar, R. (2004). "Linguistic and Statitstical Approaches to Basque Term Extraction." In: *GLAT-2004: The Production of Specialized Texts*. ISBN: 2-908849-14-3.

Anderson, J. R. (1988). "The Expert Module." In: *Foundations of Intelligent Tutoring Systems*. Ed. by M. C. Polson and J. J. Richardson. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc., pp. 21–54.

Arruarte, A. (1998). "Fundamentos y diseño de IRIS: Un entorno para la generación de Sistemas de Enseñanza Inteligentes." Doctoral dissertation. Universidad del País Vasco/Euskal Herriko Unibertsitatea (UPV/EHU).

Arruarte, A., Elorriaga, J. A., Calvo, I., Larrañaga, M., and Rueda, U. (2012). "Computer-based Concept Maps for Enabling Multilingual Education in Computer Science: a Basque, English and Spanish Languages Case." In: *Australasian Journal of Educational Technology* 28.5, pp. 793–808. ISSN: 1449-3098. URL: http://www.ascilite.org.au/ajet/ajet28/arruarte.html (visited on 04/08/2015).

Arruarte, A., Ferrero, B., Fernández de Castro, I., Urretavizcaya, M., Álvarez, A., and Greer, J. E. (2003). "The IRIS Authoring Tool." In: *Authoring Tools for Advanced Technology Learning Environments*. Ed. by T. Murray, S. Blessing, and S. Ainsworth. Kluwer Academic Publishers, pp. 233–268. ISBN: 1-4020-1772-3.

Ausubel, D. P. (1968). *Educational psychology: a cognitive view*. en. Holt, Rinehart and Winston. ISBN: 9780030696404.

Ausubel, D. P., Novak, J. D., and Hanesian, H. (1978). *Educational psychology: a cognitive view*. English. New York: Holt, Rinehart and Winston. ISBN: 0030899516 9780030899515.

El-Beltagy, S. R. and Rafea, A. (2009). "KP-Miner: A keyphrase extraction system for English and Arabic documents." In: *Information Systems* 34.1, pp. 132–144. ISSN: 0306-4379.

Benveniste, É. (1966). *Problèmes de linguistique générale*. Gallimard.

Berners-Lee, T. and Fischetti, M. (1999). *Weaving the Web: The Past, Present and Futeure of the World Wide Web by its Inventor*. Harper Business. ISBN: 0062515861.

Berners-Lee, T., Hendler, J. A., and Lassila, O. (2001). "The Semantic Web." In: *Scientific American*.

Bird, S., Klein, E., and Loper, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. Beijing: O'Reilly. ISBN: 978-0-596-51649-9.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). "DBpedia - A crystallization point for the Web of Data." In: *Journal of Web Semantics* 7.3, pp. 154–165. ISSN: 1570-8268.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). "Freebase: a collaboratively created graph database for structuring human knowledge." In: *SIGMOD '08: Proceedings of the 2008 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, pp. 1247–1250.

Bond, F. and Foster, R. (2013). "Linking and Extending an Open Multilingual Wordnet." In: *ACL (1)*. The Association for Computer Linguistics, pp. 1352–1362. ISBN: 978-1-937284-50-3.

Borst, W. N. (1997). "Construction of Engineering Ontologies for Knowledge Sharing and Reuse." Doctoral dissertation. University of Twente.

Bourdeau, J., Mizoguchi, R., Psyché, V., and Nkambou, R. (2004). "Selecting Theories in an Ontology-Based ITS Authoring Environment." In: *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004*. Ed. by J. C. Lester, R. M. Vicari, and F. Paraguaçu. Vol. 3220. Lecture Notes in Computer Science. Maceiò, Alagoas, Brazil: Springer, pp. 150–161. ISBN: 3-540-22948-5.

Bourigault, D. (1996). "Lexter, a Natural Language Processing Tool for Terminology Extraction." In: *7th EURALEX International Congress*. Goteborg.

Buitelaar, P., Cimiano, P., and Magnini (2005). "Ontology Learning from Text: An Overview." In: *Ontology Learning from Text: Methods, Applications and Evaluation*. Vol. 123. IOS Press, pp. 3–12.

Calvo, I., Arruarte, A., Elorriaga, J. A., Larrañaga, M., and Mccalla, G. (2013). "Identifying Meaningful Concept Map Elements from a Cultural Perspective." In: IEEE, pp. 250–252. ISBN: 978-0-7695-5009-1. DOI: 10.1109/ICALT.2013.77. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6601920 (visited on 12/04/2015).

Cardinaels, K. (2007). "A Dynamic Learning Object Life Cycle and its Implications for Automatic Metadata Generation." Doctoral dissertation. Faculteit Ingenieurswetenschappen, Katholieke Universiteit Leuven.

Cardinaels, K., Meire, M., and Duval, E. (2005). "Automating Metadata Generation: the Simple Indexing Interface." In: *Proceedings of the 14th International Conference on World Wide Web, WWW 2005*. Ed. by A. Ellis and T. Hagino. Chiba, Japan: ACM. ISBN: 1-59593-046-9.

Casey, J. and McAlpine, M. (2003). *Writing and Using Reusable Educational Material - A Beginner's guide*. http://assessment.cetis.ac.uk/groups/20010809144711/FR20020507112554.

Cassel, L. N., Davies, G., LeBlanc, R., Snyder, L., and Topi, H. (2008). "Using a Computing Ontology as a Foundation for Curriculum Development." In: *SW-EL'08: Ontologies and Semantic Web for Intelligent Educational Systems at 9th*

International Conference on Intelligent Tutoring Systems, ITS 2008. Ed. by D. Dicheva, A. Harrer, and R. Mizoguchi. Montreal, Canada.

Chandrasekaran, B., Josephson, J. R., and Benjamins, R. V. (1999). "What Are Ontologies, and Why Do We Need Them?" In: *IEEE Intelligent Systems* 14.1, pp. 20–26.

Cheng, X. and Roth, D. (2013). "Relational Inference for Wikification." In: *Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing, EMNLP '13*, pp. 1787–1796.

Cimiano, Philipp (2014). "Perspectives on Ontology Learning: Foreword." In: *Perspectives on Ontology Learning*. Ed. by J. Lehmann and J. Voelker. IOS Press, pp. 1–8.

Conde, A., Larrañaga, M., Arruarte, A., and Elorriaga, J. A. (submitted). "LiReWi: a Combined Method for Pedagogical Relationship Extraction from Textbooks." In: *Knowledge-based Systems*.

Conde, A., Larrañaga, M., Arruarte, A., Elorriaga, J. A., and Roth, D. (2015). "LiTeWi: A Combined Term Extraction Method for Eliciting Educational Ontologies from Textbooks." In: *Journal of the Association for Information Science Science and Technology*.

Conde, A., Larrañaga, M., Calvo, I., Arruarte, A., and Elorriaga, J. A. (2012). "Automating the Authoring of Learning Material in Computer Engineering Education." In: *Proceedings of 2012 Frontier in Education Conference*. Seattle, USA, pp. 1376–138.

Conde, A., Larrañaga, M., Lasa, A. A., and Elorriaga, J. A. (2014). "Testing Language Independence in the Semiautomatic Construction of Educational Ontologies." In: *Proceedings of 12th International Conference on Intelligent Tutoring Systems, ITS 2014*, pp. 545–550. DOI: 10.1007/978-3-319-07221-0_69. URL: http://dx.doi.org/10.1007/978-3-319-07221-0_69.

Dagan, I. and Church, K. (1994). "Termight: Identifying and Translating Technical Terminology." In: *Proceedings of the Fourth Conference on Applied Natural Language Processing*. ANLC '94. Stuttgart, Germany: Association for Computational Linguistics, pp. 34–40.

Daille, B. (1994). "Approach mixte pour l'extraction de termilogie: statistique lexicale et filters linguistiques." Doctoral dissertation. Paris: Universitè Paris VII.

Daille, B., Habert, B., Jacquemin, C., and Royauté, J. (1996). "Empirical observation of term variations and principles for their description." In: *Terminology* 3.2, pp. 197–257. ISSN: 09299971. (Visited on 03/18/2013).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). "Indexing by latent semantic analysis." In: *Journal of the American Society for Information Science* 41.6, pp. 391–407.

Dicheva, D., Sosnovsky, S. A., Gavrilova, T., and Brusilovsky, P. (2005). "Ontological Web Portal for Educational Ontologies." In: *SW-EL'05: International Workshop on Applications of the Semantic Web for E-Learning Technologies at the 12th International Conference on Artificial Intelligence in Education, AIED 2005*. Ed. by L. Aroyo and D. Dicheva. SW-EL. Amsterdam, The Netherlands, pp. 19–28. URL: `http://www.win.tue.nl/SW-EL/2005/swel05-aied05/proceedings/4-Dicheva-final-full.pdf`.

Downes, S. (2003). "Design and Reusability of Learning Objects in an Academic Context: A new Economy of Education?" In: *UDSLA Journal* 17.1.

Dunican, R. (2013). "Wikipedia & Education: Adventures in Knowledge Creation and Sharing." In: *Proceedings of the California Conference on Library Instruction.*

Dunning, T. (1993). "Accurate methods for the statistics of surprise and coincidence." In: *Comput. Linguist.* 19.1, pp. 61–74. ISSN: 0891-2017.

Earl, L. L. (1970). "Experiments in automatic extracting and indexing." In: *Information Storage and Retrieval* 6.4, pp. 313–330. ISSN: 00200271.

Elorriaga, J., Arruarte, A., Calvo, I., Larrañaga, M., Rueda, U., and Herrán, E. (2013). "Collaborative concept mapping activities in a classroom scenario." en. In: *Behaviour & Information Technology* 32.12, pp. 1292–1304. ISSN: 0144-929X, 1362-3001. DOI: `10.1080/0144929X.2011.632649`. URL: `http://www.tandfonline.com/doi/abs/10.1080/0144929X.2011.632649` (visited on 04/08/2015).

Enguehard, C. and Pantera, L. (1995). "Automatic natural acquisition of a terminology." In: *Journal of Quantitative Lingufistics* 2.1, pp. 27–32. ISSN: 0929-6174, 1744-5035.

Fano, R. (1961). *Transmission of Information: A Statistical Theory of Communications.* Cambridge, MA: The MIT Press.

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* MIT Press.

Fernando, S. (2013). "Enriching Lexical Knowledge Bases with Encyclopedic Relations." Doctoral dissertation. University of Sheffield.

Flati, T., Vannella, D., Pasini, T., and Navigli, R. (2014). *Two Is Bigger (and Better) Than One: the Wikipedia Bitaxonomy Project.* Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Baltimore, Maryland: Association for Computational Linguistics.

Fleiss, J. L. (1971). "Measuring nominal scale agreement among many raters." In: *Psychological bulletin* 76.5, p. 378.

Fok, A. W. P. and Ip, H. H. S. (2007). “Educational Ontologies Construction for Personalized Learning on the Web.” In: *Evolution of Teaching and Learning Paradigms in Intelligent Environment*. Vol. 62. SCI, pp. 47–82.

Fox, C. (1990). “A Stop List for General Text.” In: *SIGIR Forum* 24, pp. 19–35. ISSN: 0163-5840.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., and al., et (1999). “Domain-specific keyphrase extraction.” In: *Proceedings of the 16th International Joint Conference on Artificial Intelligence*. Morgan Kaufmann Publishers, pp. 668–673.

Frantzi, K., Ananiadou, S., and Mima, H. (2000). “Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method.” In: *International Journal on Digital Libraries* 3.2, pp. 115–130.

Ganapathi, G., Lourdusamy, R., and Rajaram, V. (2011). “Towards Ontology Development for Teaching Programming Language.” In: *Lecture Notes in Engineering and Computer Science* 2192.

Gavrilova, T., Farzan, R., and Brusilovsky, P. (2005). “One practical algorithm of creating teaching ontologies.” In: *Proceedings of Network Based Education*.

Gruber, T. R. (1991). “The Role of Common Ontology in Achieving Sharable, Reusable Knowledge Bases.” In: *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning, KR’91*. Morgan Kauffmann Publishers, Inc., pp. 601–602.

Guha, R. and Lenat, D. (1990). “Cyc: A Midterm Report.” In: *AI Magazine*.

HaCohen-Kerner, Y., Gross, Z., and Masa, A. (2005). “Automatic Extraction and Learning of Keyphrases from Scientific Articles.” In: *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*. Ed. by A. Gelbukh. Vol. 3406. Lecture Notes in Computer Science. Springer, pp. 657–669. ISBN: 978-3-540-24523-0. DOI: 10.1007/978-3-540-30586-6_74. URL: http://dx.doi.org/10.1007/978-3-540-30586-6%5C_74.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). “The WEKA data mining software: an update.” In: *SIGKDD Explor. Newsl.* 11.1, pp. 10–18. ISSN: 1931-0145.

Hartmann, S., Szarvas, G., and Gurevych, I. (2012). “Mining Multiwords Terms from Wikipedia.” In: *Semi-Automatic Ontology Development*, p. 226. ISBN: 9781466601888.

Hearst, M. A. (1992). “Automatic Acquisition of Hyponyms from Large Text Corpora.” In: *In Proceedings of the 14th International Conference on Computational Linguistics*, pp. 539–545.

Helmert, F. (1876). “Ueber die Wahrscheinlichkeit der Potenzsummen der Beobachtungsfehler und über einige damit im Zusammenhange stehende Fragen.” In: pp. 102–219.

Hoffart, J., Suchanek, F. M., Berberich, K., Lewis-Kelham, E., Melo, G. de, and Weikum, G. (2011). "YAGO2: exploring and querying world knowledge in time, space, context, and many languages." In: *Proceedings of the 20th international conference companion on World wide web*. New York, NY, USA: ACM, pp. 229–232.

Hoffart, J., Suchanek, F. M., Berberich, K., and Weikum, G. (2013). "YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia." In: *Artif. Intell.* 194, pp. 28–61. ISSN: 0004-3702. DOI: 10.1016/j.artint.2012.06.001. URL: http://dx.doi.org/10.1016/j.artint.2012.06.001.

Hoog, R. de, Barnard, Y., and Wielinga, B. J. (1999). "IMAT: Re-using Multi-media Electronic Technical Documentation for Training." In: *Business and Work in the Information Society: New Technologies and Applications*. Ed. by J.-Y. Roger, B. Stanford-Smith, and P. T. Kidd. IOS Press, pp. 415–421. ISBN: 90-5199-491-5.

Hulth, A. (2003). "Improved Automatic Keyword Extraction Given More Linguistic Knowledge." In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. EMNLP '03. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 216–223.

Jain, S. and Pareek, J. (2012). "Automatic Identification of Learning Object Type." In: *nternational Journal of Computer Application* 42.7, pp. 7–10.

Jiang, L., Yang, Z., Liu, Q., and Zhao, C. (2008). "The use of concept maps in educational ontology development for computer networks." In: *Granular Computing, 2008. GrC 2008. IEEE International Conference on*, pp. 346–349.

Justeson, J. S. and Katz, S. M. (1995). "Technical Terminology: Some Linguistic Properties and an Algorithm for Identification of Terms in Text." In: *Natural Language Engineering* 1.1, pp. 9–27.

Kabel, S. C., Hoog, R. de, Wielinga, B. J., and Anjewierden, A. (2004). "Indexing Learning Objects: Vocabularies and Empirical Investigation of Consistency." In: *Journal of Educational Multimedia and Hypermedia* 13.4, pp. 405–425.

Kageura, K. and Umino, B. (1996). "Methods of automatic term recognition: A review." In: *Terminology* 3.2, pp. 259–289. (Visited on 03/16/2013).

Karlsson, F., Voutilainen, A., Heikkila, J., and Anttila, A., eds. (1995). *Constraint Grammar: Language-independent System for Parsing Unrestricted Text*. Mouton de Gruyter.

Kay, J. (1999). "Ontologies for Reusable and Scrutable Student Models." In: *Workshop on Ontologies for Intelligent Educational Systems at 9th International Conference on Artificial Intelligence in Education, AI-ED' 99*. Le Mans, France.

Kay, J. and Lum, A. (2004). "Ontologies for Scrutable Student Modelling in Adaptive E-Learning." In: *SW-EL'04: Semantic Web for E-Learning. Applications of*

Semantic Web Technologies for Educational Adaptive Hypermedia Workshop at International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2004. Ed. by L. Aroyo, D. Dicheva, P. Dolog, and M. Wolpers. Eindhoven, The Netherlands.

Koilada, N., Newman, D., Lau, J. H., and Baldwin, T. (2012). “Bayesian Text Segmentation for Index Term Identification and Keyphrase Extraction.” In: Mumbai, India.

Konieczny, P. (2012). “Wikis and Wikipedia as a teaching tool: Five years later.” In: *First Monday* 19.9.

Kupiec, J., Pedersen, J., and Chen, F. (1995). “A Trainable Document Summarizer.” In: *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. SIGIR ’95. Seattle, Washington, USA: ACM, pp. 68–73. ISBN: 0-89791-714-6. DOI: 10.1145/215206.215333.

Landis, J. R. and Koch, G. G. (1977). “The measurement of observer agreement for categorical data.” In: *biometrics*, pp. 159–174.

Larrañaga, M. (2012). “Semi-automatic Generation of Learning Domain Modules for Technology Supported Learning Systems.” Doctoral dissertation. University of the Basque Country.

Larrañaga, M. (2012). “Semi-Automatic Generation of Learning Domain Modules for Technology Supported Learning Systems using Natural Language Processing Techniques and Ontologies.” Doctoral dissertation. University of the Basque Country (UPV/EHU).

Larrañaga, M., Calvo, I., Elorriaga, J. A., Arruarte, A., Verbert, K., and Duval, E. (2011). “ErauzOnt: A Framework for Gathering Learning Objects from Electronic Documents.” In: *Proceedings of the 11th IEEE International Conference on Advanced Learning Technologies, ICALT 2011*. Athens, Georgia, USA: IEEE Computer Society, pp. 656–658. ISBN: 978-0-7695-4346-8.

Larrañaga, M., Conde, A., Calvo, I., Arruarte, A., and Elorriaga, J. A. (2014). “Automatic Generation of the Domain Module from Electronic Textbooks. Method & Validation.” In: *IEEE Transactions on Knowledge and Data Engineering* 26.1, pp. 69–82.

Larrañaga, M., Conde, Á., Calvo, I., Arruarte, A., and Elorriaga, J. A. (2012). “Evaluating the Automatic Extraction of Learning Objects from Electronic Textbooks Using ErauzOnt.” In: *11th International Conference on Intelligent Tutoring Systems, ITS 2012*. Ed. by S. Cerri, W. Clancey, G. Papadourakis, and K. Panourgia. Vol. 7315. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 655–656. ISBN: 978-3-642-30949-6.

Larrañaga, M., Rueda, U., Elorriaga, J. A., and Arruarte, A. (2004). "Acquisition of the Domain Structure from Document Indexes Using Heuristic Reasoning." In: *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004*. Ed. by J. C. Lester, R. M. Vicari, and F. Paraguaçu. Vol. 3220. Lecture Notes in Computer Science. Maceiò, Alagoas, Brazil: Springer, pp. 175–186. ISBN: 3-540-22948-5.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Kleef, P. v., Auer, S., and Bizer, C. (2014). "DBpedia - A Large-scale, Multilingual Knowledge Base Extracted from Wikipedia." In: *Semantic Web Journal*.

Leidig, T. (2001). "L3–Towards an Open Learning Environment." In: *ACM Journal of Educational Resources in Computing* 1.1, pp. 5–11.

Lentini, M., Nardi, D., and Simonetta, A. (2000). "Self-instructive spreadsheets: an environment for automatic knowledge acquisition and tutor generation." In: *International Journal on Human-Computer Studies* 52.5, pp. 775–803.

Lester, J. C., Vicari, R. M., and Paraguaçu, F., eds. (2004). *Proceedings of the 7th International Conference on Intelligent Tutoring Systems, ITS 2004*. Vol. 3220. Lecture Notes in Computer Science. Maceiò, Alagoas, Brazil: Springer. ISBN: 3-540-22948-5.

Li, J., Zhao, Y., and Liu, B. (2009). "Fully Automatic Text Categorization by Exploiting WordNet." English. In: *Information Retrieval Technology*. Ed. by G. Lee, D. Song, C.-Y. Lin, A. Aizawa, K. Kuriyama, M. Yoshioka, and T. Sakai. Vol. 5839. Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 1–12. ISBN: 978-3-642-04768-8. URL: `http://dx.doi.org/10.1007/978-3-642-04769-5_1`.

Liang, C., Wu, Z., Huang, W., and Giles, C. L. (2015). "Measuring Prerequisite Relations Among Concepts." In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Lisbon, Portugal: Association for Computational Linguistics, pp. 1668–1674. URL: `http://aclweb.org/anthology/D15-1193`.

Lim, S. (2009). "How and why do college students use Wikipedia?" In: *Journal of the American Society for Information Science and Technology* 60.11, pp. 189–202.

LTSC (2001). *1484.12.1 IEEE LTSC Draft Standard for Learning Object Metadata*. URL: `http://ltsc.ieee.org/wg12/files/LOM%5C_1484%5C_12%5C_1%5C_v1%5C_Final%5C_Draft.pdf`.

Lu, R., Cao, C., Chen, Y., and Han, Z. (1995). "On Automatic Generation of Intelligent Tutoring Systems." In: *Proceedings of the 7th International Conference on Artificial Intelligence in Education, AIED 1995*. Ed. by J. E. Greer. Washington, DC: AACE, pp. 67–74.

Maedche, A. and Staab, S. (2000). “Mining Ontologies from Text.” In: *Proceedings of the 12th International Conference on Knowledge Engineering and Management, EKAW 2000*. Vol. 1937. Springer, pp. 189–202.

Mahdisoltani, F., Biega, J., and Suchanek, F. M. (2014). “YAGO3: A Knowledge Base from Multilingual Wikipedias.” In:

Martin, B. and Mitrovic, A. (2003). “ITS Domain Modelling: Art or Science?” In: *Shaping the Future of Learning through Intelligent Technologies, Proceedings of the 11th International Conference on Artificial Intelligence in Education, AIED 2003*. Ed. by U. Hoppe, F. Verdejo, and J. Kay. Vol. 97. Frontiers in Artificial Intelligence and Applications. Sidney, Australia: IOS Press, pp. 183–190. ISBN: 1-58603-356-6.

Maynard, D. and Ananiadou, S. (1999). “Identifying Contextual Information for Multi-Word Term Extraction.” In: *In (Sandrini)*, pp. 212–221.

Medelyan, O. and Witten, I. H. (2006). “Thesaurus Based Automatic Keyphrase Indexing.” In: *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*. JCDL ’06. Chapel Hill, NC, USA: ACM, pp. 296–297. ISBN: 1-59593-354-9. DOI: 10.1145/1141753.1141819. URL: http://doi.acm.org/10.1145/1141753.1141819.

Meire, M., Ochoa, X., and Duval, E. (2007). “SAmgI: Automatic Metadata Generation v2.0.” In: *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2007, ED-MEDIA 2007*. Ed. by C. Montgomerie and J. Seale. Vancouver, Canada: AACE, pp. 1195–1204. URL: http://www.editlib.org/p/25528.

Melo, G. de and Weikum, G. (2014). “Taxonomic data integration from multilingual Wikipedia editions.” en. In: *Knowledge and Information Systems* 39.1, pp. 1–39. ISSN: 0219-1377, 0219-3116. DOI: 10.1007/s10115-012-0597-3. URL: http://link.springer.com/10.1007/s10115-012-0597-3 (visited on 10/29/2014).

Milne, D. and Witten, I. H. (2008). “Learning to Link with Wikipedia.” In: *Proceedings of the 17th ACM Conference on Information and Knowledge Management*. CIKM ’08. Napa Valley, California, USA: ACM, pp. 509–518. ISBN: 978-1-59593-991-3.

Milne, D. and Witten, I. H. (2013). “An open-source toolkit for mining Wikipedia.” In: *Artificial Intelligence* 194, pp. 222–239. ISSN: 0004-3702.

Mima, H. and Ananiadou, S. (2000). “An Application and Evaluation of the C/NC-value Approach for the Automatic term Recognition of Multi-Word units in Japanese.” In: *International Journal of Terminology* 6.2, pp. 175–194.

Mitrovic, A., Koedinger, K. R., and Martin, B. (2003). “A Comparative Analysis of Cognitive Tutoring and Constraint-Based Modeling.” In: *User Modeling*. Ed.

by P. Brusilovsky, A. T. Corbett, and F. de Rosis. Vol. 2702. Lecture Notes in Computer Science. Johnstown, PA: Springer, pp. 313–322. ISBN: 3-540-40381-7.

Mitrovic, A., Suraweera, P., Martin, B., and Weerasinghe, A. (2004). "DB-suite: Experiences with Three Intelligent, Web-based Database Tutors." In: *Journal of Interactive Learning Research* 15.4, pp. 409–432.

Mizoguchi, R. and Bourdeau, J. (2000). "Using Ontological Engineering to Overcome AI-ED Problems." In: *International Journal on Artificial Intelligence in Education* 11.2, pp. 107–121.

Mizoguchi, R., Ikeda, M., and Sinitsa, K. (1997). "Roles of Shared Ontology in AI-ED Research." In: *Knowledge and Media in Learning Systems. Proceedings of the 8th International Conference on Artificial Intelligence in Education, AIED-1997.* Ed. by B. du Boulay and R. Mizoguchi. Vol. 39. Frontiers in Artificial Intelligence and Applications. Kobe, Japan: IOS Press, pp. 537–544. ISBN: 90-5199-353-6.

Morison, I. (2008). *Introduction to Astronomy and Cosmology.* Wiley.

Murray, T. (1998). "Authoring Knowledge-Based Tutors: Tools for Content, Instructional Strategy, Student Model and Interface Design." In: *The Journal of the Learning Sciences* 7.1, pp. 5–64.

Murray, T. (1999). "Authoring Intelligent Tutoring Systems: An analysis of the state of the art." In: *International Journal on Artificial Intelligence in Education* 10, pp. 98–129.

Murray, T. (2003). "EON: Knowledge Based Tutor Authoring Tools for Content, Instructional, Strategy, student Model and Interface Design." In: *Authoring Tools for Advanced Technology Learning Environments*. Ed. by T. Murray, S. Blessing, and S. Ainsworth. Kluwer Academic Publishers. ISBN: 1-4020-1772-3.

Murray, T., Blessing, S., and Ainsworth, S., eds. (2003). *Authoring Tools for Advanced Technology Learning Environments.* Kluwer Academic Publishers. ISBN: 1-4020-1772-3.

Nastase, V. and Strube, M. (2008). "Decoding Wikipedia Categories for Knowledge Acquisition." In: *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 2.* AAAI'08. Chicago, Illinois: AAAI Press, pp. 1219–1224. ISBN: 978-1-57735-368-3.

Navigli, R. and Ponzetto, S. P. (2012). "BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network." en. In: *Artificial Intelligence* 193, pp. 217–250. ISSN: 00043702. (Visited on 10/29/2014).

Neches, R., Fikes, R., Finin, T., Gruber, T. R., Senator, T., and Swartout, W. (1991). "Enabling Technology for Knowledge Sharing." In: *AI Magazine* 12.3, pp. 36–56.

Nkambou, R., Bourdeau, J., and Mizoguchi, R., eds. (2010). *Advances in Intelligent Tutoring Systems*. Vol. 308. Studies in Computational Intelligence. Springer. ISBN: 978-3-642-14362-5.

Novak, J. D. (1977). *A Theory of Education*. Cornell University, Ithaca, NY. Cornell University Press.

Novak, J. D. and Cañas, A. J. (2008). *The theory underlying concept maps and how to construct them*. Tech. rep. IHMC CmapTools 2006-01 Rev 01- 2008, Florida: Institute for Human and Machine Cognition. URL: `http://cmap.ihmc.us/Publications/ResearchPapers/TheoryCmaps/TheoryUnderlyingConceptMaps.bck-11-01-06.htm` (visited on 04/08/2015).

Padró, L. and Stanilovsky, E. (2012). "FreeLing 3.0: Towards Wider Multilinguality." In: *Proceedings of the Language Resources and Evaluation Conference (LREC 2012)*. Istanbul, Turkey.

Paukkeri, M.-S., Nieminen, I. T., Pöllä, M., and Honkela, T. (2008). "A Language-Independent Approach to Keyphrase Extraction and Evaluation." In: *Coling 2008: Companion volume: Posters*. Manchester, UK: Coling 2008 Organizing Committee, pp. 83–86.

Pazienza, M. T., Pennacchiotti, M., and Zanzotto, F. M. (2005). "Terminology Extraction: An Analysis of Linguistic and Statistical Approaches." In: *Knowledge Mining*. Ed. by D. S. Sirmakessis. Studies in Fuzziness and Soft Computing 185. Springer Berlin Heidelberg, pp. 255–279. ISBN: 978-3-540-25070-8, 978-3-540-32394-5.

Pazienza, M. T. and Stellato, A., eds. (2012). *Semi-automatic ontology development: processes and resources*. Hershey, PA: Information Science Reference. ISBN: 9781466601888.

Plackett, R. L. (1983). "Karl Pearson and the Chi-Squared Test." In: *International Statistical Review / Revue Internationale de Statistique* 51.1, pp. 59–72. ISSN: 03067734.

Pohl, A. (2012). "Improving the Wikipedia Miner word sense disambiguation algorithm." In: *Computer Science and Information Systems (FedCSIS), 2012 Federated Conference on*, pp. 241–248.

Ponzetto, S. P. and Strube, M. (2007). "Deriving a large scale taxonomy from Wikipedia." In: *Proceedings of the 22nd national conference on Artificial intelligence - Volume 2*. AAAI'07. AAAI Press, pp. 1440–1445. ISBN: 978-1-57735-323-2. URL: `http://dl.acm.org/citation.cfm?id=1619797.1619876`.

Porter, M. F. (1997). "Readings in Information Retrieval." In: ed. by K. Sparck Jones and P. Willett. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Chap. An Algorithm for Suffix Stripping, pp. 313–316. ISBN: 1-55860-454-5.

Raineri, D. (2010). *11th Hour: Introduction to Molecular Biology.* Wiley.

Ratinov, L. and Roth, D. (2009). "Design Challenges and Misconceptions in Named Entity Recognition." In: *CoNLL.* URL: `http://cogcomp.cs.illinois.edu/papers/RatinovRo09.pdf`.

Ratinov, L., Roth, D., Downey, D., and Anderson, M. (2011). "Local and Global Algorithms for Disambiguation to Wikipedia." In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, HLT '11*, pp. 1375–1384.

Robin, C. R. and Uma, G. V. (2011). "An ontology based linguistic infrastructure to represent software risk identification knowledge." In: *Proceedings of the International Conference & Workshop on Emerging Trends in Technology.* ACM, pp. 744–751. (Visited on 11/14/2014).

Rose, S., Engel, D., Cramer, N., and Cowley, W. (2010). "Automatic Keyword Extraction from Individual Documents." In: *Text Mining. Applications and Theory.* Ed. by M. W. Berry and J. Kogan. John Wiley and Sons, Ltd, pp. 1–20. ISBN: 9780470689646. DOI: `10.1002/9780470689646.ch1`. URL: `http://dx.doi.org/10.1002/9780470689646.ch1`.

Roy, D. (2006). "Automatic Annotation of Learning Materials for E-Learning." Doctoral dissertation. Indian Institute of Technology.

Roy, D., Sarkar, S., and Ghose, S. (2008). "Automatic Extraction of Pedagogic Metadata from Learning Content." In: *International Journal on Artificial Intelligence in Education* 18.2, pp. 97–118.

Salton, G. (1971). *The SMART Retrieval System ;Experiments in Automatic Document Processing.* Upper Saddle River, NJ, USA: Prentice-Hall, Inc.

Salton, G., Wong, A., and Yang, C. S. (1975). "A vector space model for automatic indexing." In: *Commun. ACM* 18.11, pp. 613–620. ISSN: 0001-0782. (Visited on 12/11/2012).

Salton, G. and Buckley, C. (1988). "Term-weighting approaches in automatic text retrieval." In: *Information Processing and Management*, pp. 513–523.

Silva, J. F. da and Lopes, G. P. (1999). "A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora." In: *Sixth Meeting on Mathematics of Language*, pp. 369–381.

Simon, B., Massart, D., Assche, F. V., Ternier, S., Duval, E., Brantner, S., Olmedilla, D., and Miklós, Z. (2005). "A Simple Query Interface for Interoperable Learning Repositories." In: *Proceedings of the 1st Workshop on Interoperability of Web-based Educational Systems at the 14th International World Wide Web Conference, WWW 2005.* Chiba, Japan: CEUR, pp. 11–18.

Sorg, P. and Cimiano, P. (2012). “Exploiting Wikipedia for cross-lingual and multilingual information retrieval.” In: *Data and Knowledge Engineering* 72.72, pp. 26–45.

Sosnovsky, S. and Gavrilova, T. (2006). “Development of Educational Ontology for C-programming.” In: *International Journal on Information Theories & Applications* 13.4, pp. 303–308. (Visited on 12/10/2012).

Studer, R., Benjamins, V. R., and Fensel, D. (1998). “Knowledge Engineering, Principles and Methods.” In: *Data and Knowledge Engineering* 25.1-2, pp. 161–197.

Suchanek, F. M., Kasneci, G., and Weikum, G. (2007). “Yago: A Core of Semantic Knowledge.” In: *16th international World Wide Web conference (WWW 2007).* New York, NY, USA: ACM Press.

Ternier, S., Massart, D., Assche, F. V., Smith, N., Simon, B., and Duval, E. (2008). “A Simple Publishing Interface for Learning Object Repositories.” In: *Proceedings of the World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008, ED-MEDIA 2008.* AACE. Vienna, Austria, pp. 1840–1845.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). “Feature-rich Part-of-speech Tagging with a Cyclic Dependency Network.” In: *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1.* NAACL ’03. Edmonton, Canada: Association for Computational Linguistics, pp. 173–180.

Turney, P. (2000). “Learning Algorithms for Keyphrase Extraction.” In: *Information Retrieval* 2, pp. 303–336.

Unesco (2003). *Education in a multilingual world.* Paris (France): United Nations Educational, Scientific and Cultural Organization. URL: `http://unesdoc.unesco.org/images/0012/001297/129728e.pdf`.

Uschold, M. and Gruninger, M. (1996). “Ontologies: Principles, Methods and Applications.” In: *Knowledge Engineering Review* 11.2, pp. 93–136.

Velardi, P., Missikoff, M., and Basili, R. (2001). “Identification of Relevant Terms to Support the Construction of Domain Ontologies.” In: *Proceedings of the Workshop on Human Language Technology and Knowledge Management - Volume 2001.* HLTKM ’01. Stroudsburg, PA, USA: Association for Computational Linguistics, 5:1–5:8.

Verbert, K., Gašević, D., Jovanović, J., and Duval, E. (2005). “Ontology-Based Learning Content Repurposing.” In: *Proceedings of the 14th International Conference on World Wide Web, WWW 2005 - Special interest tracks and posters.* Ed. by A. Ellis and T. Hagino. Chiba, Japan: ACM Press, pp. 1140–1141. ISBN: 1-59593-051-5.

Verbert, K., Ochoa, X., and Duval, E. (2008). "The ALOCOM Framework: Towards Scalable Content Reuse." In: *Journal of Digital Information* 9.1.

Verbert, K. (2008). "An Architecture and Framework for Flexible Reuse of Learning Object Components." Doctoral dissertation. Leuven, Belgium: Faculteit Ingenieurswetenschappen, Katholieke Universiteit Leuven. ISBN: 978-90-5682-905-6.

Vrandečić, D. and Krötzsch, M. (2014). "Wikidata: A Free Collaborative Knowledgebase." In: *Commun. ACM* 57.10, pp. 78–85. ISSN: 0001-0782. DOI: `10.1145/2629489`. URL: `http://doi.acm.org/10.1145/2629489`.

Wang, S., Liang, C., Wu, Z., Williams, K., Pursel, B., Bräutigam, B., Saul, S., Williams, H., Bowen, K., and Giles, C. L. (2015). "Concept Hierarchy Extraction from Textbooks." In: *DocEng*. Ed. by C. Vanoirbeek and P. Genevés. ACM, pp. 147–156. ISBN: 978-1-4503-3307-8. URL: `http://dblp.uni-trier.de/db/conf/doceng/doceng2015.html#WangLWWPBSWBG15`.

Wenger, E. (1987). "Artificial Intelligence and Tutoring Systems. Computational and Cognitive Approaches to the Communication of Knowledge." In: *Artificial Intelligence and Tutoring Systems*. Morgan Kauffmann Publishers, Inc. ISBN: 0-934613-26-5.

Westerhout, E. (2009). "Definition Extraction Using Linguistic and Structural Features." In: *Proceedings of the 1st Workshop on Definition Extraction*. WDE '09. Borovets, Bulgaria: Association for Computational Linguistics, pp. 61–67. ISBN: 978-954-452-013-7. URL: `http://dl.acm.org/citation.cfm?id=1859765.1859775`.

White, T. (2010). *Hadoop: The Definitive Guide*. Second. O'Reilly Media.

Whitley, D. (1989). "The GENITOR Algorithm and Selection Pressure: Why Rank-based Allocation of Reproductive Trials is Best." In: *Proceedings of the Third International Conference on Genetic Algorithms*. George Mason University, USA: Morgan Kaufmann Publishers Inc., pp. 116–121. ISBN: 1-55860-006-3.

Wong, S. and Nguyen, D. (2010). *Principles of Object-Oriented Programming*. URL: `http://cnx.org/content/col10213/1.37/`.

Woolf, B. P. (2008). *Building Intelligent Interactive Tutors: Student-centered Strategies for Revolutioniizing E-learning*. Morgan Kauffmann Publishers, Inc. ISBN: 978-0123735942.

Wu, F. and Weld, D. S. (2008). "Automatically refining the Wikipedia infobox ontology." In: ACM Press, p. 635. ISBN: 9781605580852. (Visited on 09/17/2013).

Zhang, Z., Brewster, C., and Ciravegna, F. (2008). "Ciravegna F: A Comparative Evaluation of Term Recognition Algorithms." In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC08)*. Marrakech, Morocco 2008.

Zouaq, A. and Nkambou, R. (2009). “Enhancing Learning Objects with an Ontology-Based Memory.” In: *IEEE Transactions on Knowledge and Data Engineering* 21.6, pp. 881–893.

Appendices

A

Spanish Trainning Data for Wikiminer Relatedness Classifier

In this appendix, the data used to train the Wikiminer relatedness classifier can be seen (Table A.1). The table shows the different term pairs used along with their Wikipedia mappings and their assigned relatedness score.

Topic 1	Wikipedia Id.	Topic 2	Wikipedia Id.	Score
Amor	23565	Sexo	4169616	6.77
Tigre	25113	Gato	409000	7.35
Tigre	25113	Tigre	25113	10
Libro	21933	Papel	12349	7.46
Ordenador	8985	Teclado	510471	7.62
Ordenador	8985	Internet	4507055	7.58
Avion	466366	Coche	18019	5.77
Tren	5428	Coche	18019	6.31
Telefono	6257	Comunicación	8605	7.5
Televisión	5693	Radio	2226582	6.77
Media	1133166	Radio	2226582	7.42
Droga	2471190	Abuso	15911	6.85
Pan	28138	Mantequilla	55812	6.19
Cucumis sativus	89089	Patata	15162	5.92
Médico	2564590	Enfermera	230857	7
profesor	79570	Doctor	87496	6.62
Estudiante	199231	Profesor	79570	6.81
Empresa	1586816	Acción (finazas)	121510	7.08
Acción	121510	Mercado	14762	8.08
Acción	121510	Teléfono	30003	1.62
Acción	121510	CD	4888	1.31
Acción	121510	Jaguar Cars	158259	0.92

Topic 1	Wikipedia Id.	Topic 2	Wikipedia Id.	Score
Fondo (caldo)	5016233	Huevo (alimento)	138544	1.81
Fertilidad	27284	Huevo (biología)	1208111	6.69
Libro	21933	Biblioteca	402	7.46
Banco	42950	Dinero	23556	8.12
Madera	2654056	Bosque	19291	7.73
Dinero	23556	Dinero en efectivo	278165	9.15
Profesor	79570	Cucumis sativus	89089	0.31
Monarca	28253	Berza (fruta)	1416906	0.23
Monarca	28253	Reina	40343	8.58
Rey (ajedrez)	94056	Torre (ajedrez)	94054	5.92
Obispo	638456	Rabino	72658	6.69
Jerusalén	30029	Israel	10005	8.46
Jerusalén	30029	Pueblo palestino	3125108	7.65
Sagrado	181093	Sexo	4169616	1.62
Maradona	9496	Fútbol	674732	8.62
Fútbol	674732	Gútbol	674732	9.03
Fútbol	674732	Baloncesto	366	6.81
Fútbol	674732	Tenis	5340	6.63
Tenis	5340	Raqueta	250868	7.56
Yasir Arafat	22890	Paz	43655	6.73
Derecho	860	Abogado	54263	8.38
Ppelícula	26900	Celebridad	776488	7.38
Película	26900	Palomitas de maíz	86407	6.19
Película	26900	Crítica	52420	6.73
Película	26900	Sala de proyección	487655	7.92
Física	1155	Protón	2139	8.12
Física	1155	Química	3354	7.35
Espacio	4073767	Química	3354	4.88
Alcohol	7613	Química	3354	5.54
Vodka	87725	Ginebra (bebida)	594920	8.46
Vodka	87725	Brandy	150546	8.13
Bebida	15998	Coche	18019	3.04
Bebida	15998	Oído	45403	1.31
Bebida	15998	Boca	89887	5.96
Bebida	15998	Alimentaci'on	19324	6.87
Bebé	47351	Madre	204484	7.85
Bebida	15998	Madre	204484	2.65
coche	18019	Automóvil	18019	8.94
gema	11607	Gema	11607	8.96
Costa	89250	Ribera (orilla)	199696	9.1
Hospital psiqui'atrico	689261	Casa de locos	689261	8.87
Mago	27550	Ilusionista	83179	9.02
MediodÃŋa	58912	Mediodía	58912	9.29
Estufa (calefacci'on)	2054257	Estufa	2054257	8.79

Topic 1	Wikipedia Id.	Topic 2	Wikipedia Id.	Score
Alimento	148	Fruto	31571	7.52
Aves	906733	Gallo	1409670	7.1
Aves	906733	Gruidae	69602	7.38
Hermano	534194	Monje	445853	6.27
Viaje	2043985	Coche	18019	5.85
Monje	445853	Oraculo	651827	5
Cementerio	84163	Bosque	19291	2.08
Alimento	148	Gallo	1409670	4.42
Costa	89250	Colina	337965	4.38
Bosque	19291	Cementerio	84163	1.85
Ribera (orilla)	199696	Bosque	19291	3.08
Monje	445853	Esclavo	43139	0.92
Costa	89250	Costa	89250	3.15
Cuerda (geometría)	5301606	Sonrisa	141352	0.54
Vidrio	26553	Mago	27550	2.08
Mediodía	58912	Cordel (textil)	3133413	0.54
Gallo	1409670	Viaje	2043985	0.62
Dinero	23556	Dólar	21288	8.42
Dinero	23556	Dinero en efectivo	278165	9.08
Dinero	23556	Divisa	2476004	9.04
Dinero	23556	Riqueza	160093	8.27
Dinero	23556	Propiedad	182912	7.57
Dinero	23556	Posesión	130204	7.29
Dinero	23556	Banco	42950	8.5
Dinero	23556	Cuenta de ahorro	1248594	7.73
Dinero	23556	Jubilación	360707	6.88
Dinero	23556	Lavado de dinero	1884088	5.65
Dinero	23556	Cirugía	1529521	3.31
Tigre	25113	Panthera onca	26796	8
Tigre	25113	Felidae(felino)	31555	8
Tigre	25113	Carnívoro	18053	7.08
Tigre	25113	Mammalia(mamifero	26079	6.85
Tigre	25113	Animalia(animales)	235	7
Tigre	25113	Ser vivo	7624	4.77
Tigre	25113	Fauna	34684	5.62
Tigre	25113	Zoo	28250	5.87
Psicología	2127	Psiquiatría	12675	8.08
Psicología	2127	Trastorno de ansiedad	5204342	7
Psicología	2127	Miedo	72322	6.85
Psicología	2127	Depresión	29840	7.42
Psicología	2127	Clínica	40939	6.58
Psicología	2127	Médico	2564590	6.42
Psicología	2127	Sigmund Freud	19304	8.21
Psicología	2127	Mente	141256	7.69

Topic 1	Wikipedia Id.	Topic 2	Wikipedia Id.	Score
Psicología	2127	Salud	2680	7.23
Psicología	2127	Ciencia	622	6.71
Psicología	2127	Disciplina	2429965	5.58
Psicología	2127	Cognición	4201688	7.48
Planeta	2141	Estrella	3364	8.45
Planeta	2141	Constelación	441	8.06
Planeta	2141	Luna	5189863	8.08
Planeta	2141	Sol	2570	8.02
Planeta	2141	Galaxia	7364	8.11
Planeta	2141	Espacio (física)	4073767	7.92
Planeta	2141	Astrónomo	46117	7.94
Stare decisis	440171	Información	9718	3.85
Stare decisis	440171	Cognición	4201688	2.81
Stare decisis	440171	Derecho	860	6.65
Stare decisis	440171	Colección (conjunto)	806	2.5
Stare decisis	440171	Grupo	32419	1.77
Taza	509486	Café	298700	6.58
Taza	509486	Vajilla	415314	6.85
Taza	509486	Artefacto arqueológico	4462580	2.92
Taza	509486	Entidad	84060	2.15
Taza	509486	Bebida	15998	7.25
Taza	509486	Alimento	148	5
Taza	509486	Substancia	909196	1.92
Taza	509486	Líquido	698355	5.9
Panthera onca	26796	Gato	409000	7.42
Jaguar Cars	158259	Automóvil	18019	7.27
Energía	2482150	Secretario	393541	1.81
Energía	2482150	Laboratorio	1832489	5.09
Computadora	8985	Laboratorio	1832489	6.78
Arma	16080	Secreto	683381	6.06
FBI	54927	Huella dactilar	195819	6.94
Marte (planeta)	1787	Agua	2444645	2.94
Marte (planeta)	1787	Científico	13261	5.63
Noticia	85311	Informe	1008058	8.16
Cañón (geomorfología)	55751	Paisaje	3468	7.53
Imagen	4316322	Superficie (matemática)	4218	4.56
Transbordador espacial	13607	Espacio (física)	4073767	6.34
Agua	2444645	Mecánica de suelos	154344	6.56
Símbolo monetario	1405324	Recreo estudiantil	4869561	2.38
MiÃľrcoles	11148	Noticia	85311	2.22
Milla	1766	Kilómetro	1597	8.66
Computadora	8985	Noticia	85311	4.47
Territorio	956936	Superficie (matemática)	4218	5.34
Atmósfera	208166	Paisaje	3468	3.69

Topic 1	Wikipedia Id.	Topic 2	Wikipedia Id.	Score
Presidente USA	28105	Medalla	148402	3
Piel	6626	Ojo	3172583	6.22
Japoneés (etnia)	1328767	Estados Unidos	2400722	6.5
Teatro	2780	Historia	1370	3.91
Voluntariado	176743	Lema	1269442	2.56
Siglo	21412	Año	37	7.59
Siglo	21412	Nación	666600	3.16
Ministro	89618	Partido político	27039	6.63
Minoría	525816	Paz	43655	3.69
Gobierno	1314	Crisis	394822	6.56
Energía	2482150	Crisis	394822	5.94
Accidente cerebrovascular	41549	Hospital	30364	7.03
Discapacidad	29158	Muerte	1942	5.47

Table A.1 – Wikiminer Training Data

B

Stopword Lists

In this Appendix the different stopword lists applied to the process of gathering the topics of the LDO *LiTeWi* (Chapter 4) are shown.

B.1 Stopword List Applied to TF-IDF Index

"a", "an", "and", "are", "as", "at", "be", "but", "by", "for", "if", "in", "into", "is", "it", "no", "not", "of", "on", "or", "such", "that", "the", "their", "then", "there", "these", "they", "this", "to", "was", "will", "with".

B.2 Stopword List Applied to the Term Extraction Techniques

"a", "about", "above", "after", "again", "against", "all", "am", "an", "and" "any", "are", "aren’t", "as", "at", "be", "because", "been", "before", "being", "below", "between", "both", "but", "by", "can’t", "cannot", "could", "couldn’t", "did", "didn’t", "do", "does", "doesn’t", "doing", "don’t", "down", "during", "each", "few", "for", "from", "further", "had", "hadn’t", "has", "hasn’t", "have", "haven’t", "having", "he", "he’d", "he’ll", "he’s", "her", "here", "here’s", "hers", "herself", "him", "himself", "his", "how", "how’s", "i", "i’d", "i’ll", "i’m", "i’ve", "if", "in", "into", "is", "isn’t", "it", "it’s", "its", "itself", "let’s", "me", "more", "most", "mustn’t", "my", "myself", "no", "nor", "not" , "of", "off", "on", "once", "only", "or", "other", "ought", "our", "ours", "ourselves", "out", "over", "own", "same", "shan’t", "she", "she’d", "she’ll", "she’s", "should", "shouldn’t", "so", "some", "such", "than", "that", "that’s", "the", "their", "theirs", "them", "themselves", "then", "there", "there’s", "these", "they", "they’d", "they’ll", "they’re", "they’ve", "this", "those", "through", "to", "too", "under", "until", "up", "very", "was", "wasn’t", "we", "we’d" ,"we’ll" ,"we’re" ,"we’ve" , "were", "weren’t" ,"what", "what’s", "when", "when’s", "where", "where’s", "which", "while", "who", "who’s", "whom", "why", "why’s", "with", "won’t", "would", "wouldn’t", "you", "you’d", "you’ll", "you’re", "you’ve", "your", "yours", "yourself", "yourselves", "*", "/", "!", "?", "a", "able", "about", "across", "after", "all", "almost", "also", "am, "among", "an", "and", "any", "are", "as", "at", "be", "because", "been", "but", "by", "can", "cannot", "could","dear ","did","do", "does", "either", "else", "ever", "every", "for", "from", "get", "got", "had", "has", "have", "he", "her", "hers", "him", "his", "how", "however", "i", "if", "in" ,"into", "is", "it", "its", "just", "least", "let", "like", "likely", "may", "me","might","most", "must","my","neither", "no", "nor", "not", "of", "off", "often", "on","only","or","other", "our", "own", "rather", "said", "say" , "says", "she", "should", "since", "so", "some", "than", "that", "the", "their", "them" , "then" , "there", "they", "to", "too", "was", "us", "wants", "was", "we", "were", "what", "when", "where", "which", "while", "who", "whom", "why", "will", "with", "would", "yet", "+", "-", "[", "]", "", "", ".", ",", "(", ")", "whose", "[", ">", "etc".