USING LINGUISTIC PATTERNS TO ENHANCE ONTOLOGY DEVELOPMENT

Guadalupe Aguado de Cea, Asunción Gómez-Pérez,
Elena Montiel-Ponsoda, Mari Carmen Suárez-Figueroa
Ontology Engineering Group, Dpto. de Inteligencia Artificial, Facultad de Informática, Universidad Politécnica de Madrid
Campus de Montegancedo s/n, 28660 Boadilla del Monte, Madrid, Spain
{lupe, asun, emontiel, mcsuarez}@fi.upm.es

Keywords: ontology development, ontology design patterns, ontology statements, lexico-syntactic patterns.

Abstract: In this paper we describe how linguistic patterns can contribute to ontology development by enabling an easier reuse of some ontological resources. In particular, our research focuses on the reuse of ontology design patterns and ontology statements by relying on linguistic constructs at different stages of the reuse process. With this aim, we propose the employment of lexico-syntactic patterns with two objectives: 1) the reuse of ontology design patterns, and 2) the validation of ontology statements for their subsequent reuse in the ontology development. To illustrate the proposed approaches, we will present some examples of lexico-syntactic patterns and their employment in the reuse of ontology design patterns and in the validation of ontology statements.

1 INTRODUCTION

The 1990s and the first years of this new millennium have witnessed the growing interest of many practitioners in methodologies and methods that support ontology development. A complete overview of the methods and methodologies for developing ontologies from scratch has been reported in (Gómez-Pérez *et al.*, 2003). All these efforts have contributed to transform the art of constructing single ontologies into an engineering activity. However, in the last years, ontology developers are starting to reuse and re-engineer as much as possible knowledge-aware resources (e.g., ontologies, ontology modules, ontology statements, ontology design patterns, thesauri, lexicons, classification schemas, and databases) as opposed to custom-building ontologies from scratch. This new ontology development paradigm has the following benefits: (a) allows speeding up the ontology development process, (b) saves time and budget, and (c) promotes the application of good practices.

In this new paradigm, design patterns have appeared into scene. Design patterns are generally defined as *archetypal solutions to design problems* (Gangemi, 2005). According to (Menzies, 1997), design patterns are generally perceived as having three kinds of benefits: (1) reuse, (2) guidance, and (3) fluent communication. Intuitively, these benefits can also be applied to Ontology Engineering, and recently experiments have empirically proven this intuition (Dzbor *et al.*, 2009). On the light of these benefits, the Ontology Engineering community is devoting many efforts to the development of ontology design pattern repositories (e.g. Ontology Design Patterns.org) to encourage users the reuse of best practices while speeding up ontology development. However, experiments in the reuse of Ontology Design Patterns (ODPs hereafter) have also revealed that users with different levels of expertise in ontology modelling face difficulties when trying to reuse ODPs. The set of ODPs used in this study have been designed in the NeOn project (www.neon-project.org), and are included in (Suárez-Figueroa *et al.*, 2007).

Other ontological resources worth reusing for accelerating the ontology development process are ontology statements. We consider an ontology statement (or triple) as containing three components: *subject*, *predicate* and *object*. Such ontology statements are available in the Semantic Web (SW) and can be used to build a preliminary model of the ontology or to extend and improve an existing one. Nevertheless, an important issue here is the quality

of the retrieved ontological knowledge represented by the ontology statement.

With the aim of offering a solution for a flawless reuse of both ODPs and ontology statements, we are currently developing an innovative approach based on a Natural Language human interface. In the same way that the knowledge of the world used to build knowledge systems is normally captured in Natural Language (NL), it seems reasonable that NL be one of the most appropriate means to identify and validate encoded knowledge.

Based on previous work on knowledge extraction from texts summarized in (Cimiano, 2006), our intuition is that some linguistic constructs reliably convey ontological relations. The linguistic constructs we deal with in the present work combine lexical items with syntactic dependencies and have *verbs* as main elements. For the purposes of our research, we have adopted the term *Lexico-Syntactic Patterns* or LSPs (Hearst, 1992) to designate those linguistic constructs that have been defined in (Aguado de Cea *et al.*, 2008) as "formalized linguistic schemas or constructions derived from regular expressions in NL that consist of certain linguistic and paralinguistic elements, following a specific syntactic order, and that permit to extract some conclusions about the meaning they express".

To guarantee an easy reuse of ODPs, a repository of LSPs corresponding to ODPs has been developed and published in (Aguado de Cea *et al.*, 2008). The purpose of this repository is to enable the reuse of ODPs starting from sentences in NL formulated by users. In this paper, we will contribute with some enhancements to the LSPs repository. Regarding the reuse of ontology statements, the goal is to provide a way for predicting the truth value of these statements also by means of LSPs, with the aim of ensuring the correctness of the knowledge to be reused. Thus, when developing ontologies, LSPs will be highly useful because they facilitate 1) the reuse of the ODPs that represent the phenomenon or domain aspect the user wants to model in the ontology, and 2) the reuse of validated ontology statements drawn from online ontologies.

The reminder of this paper is structured as follows. Section 2 is devoted to the motivation. Section 3 presents some examples of LSPs and improvements to the initial repository. Section 4 includes an enhanced version of the guidelines for reusing of ODPs. Section 5 proposes some steps for the validation of ontology statements. Finally, we conclude the paper and outline some future lines of research in section 6.

2 MOTIVATION

Ontology Design Patterns Reuse. Some experiments reported in (Aguado de Cea *et al.*, 2008) showed common problems, errors and misunderstandings faced by novice users when they tried to reuse ODPs to solve modelling problems expressed in NL. The ODPs used in the mentioned experiments are included in (Suárez-Figueroa *et al.*, 2007), and have been classified into Logical and Content ODPs. Examples of Logical ODPs are the Logical Pattern for modelling *SubclassOf* relation, the one for *Universal Restrictions*, or the one for *Disjoint Classes*. Under the category of Content Patterns, we find the *Participation*, the *Role-Task*, or the *Simple Part-Whole* relation patterns. Subjects participating in the mentioned experiments had problems in correctly reusing the *Part-Whole* relation Content pattern, for example, and mistook it for the Logical ODPs for *Object Property*, *SubclassOf*, or *SubpropertyOf*. Another typical error was the incorrect use of the *Exhaustive Classes* ODP, when the *Disjoint Classes* ODP should have been employed.

Ontology Statements Validation. One crucial issue in the current SW is related with the trust assigned to the knowledge accessed through SW portals like Watson (http://watson.kmi.open.ac.uk). Several studies (Sabou *et al.*, 2008; Suárez-Figueroa *et al.*, 2008) have revealed the need for evaluating ontology statements before reusing them, because of the large number of incorrect or misleading statements retrieved from the Web. For a classification of the identified problematic statements see (Sabou *et al.*, 2009). Problematic statements that are of interest to this research are the so-called *modelling errors*, i.e., the ones that express an incorrect relation from a (formal) modelling perspective. One of the most common errors in online ontologies reported in (Sabou *et al.*, 2008) was the use of the *SubclassOf* relation (or subsumption) "as a way to model the fact that there exists some type of relation between two concepts". This resulted in examples like *Biographies subclass_of People*. Another quite common error was the use of the *SubclassOf* relation to model *Part-Whole* relations. For example, *Branch subclass_of Tree*.

The results of these experiments shed some light about incorrect modelling of some of the most common ontological relations when reusing ODPs or ontology statements. This brought us to propose a NL-based approach to facilitate the access to ODPs,

and to validate the knowledge expressed in ontology statements, since NL is the way humans have to communicate knowledge.

3 LEXICO-SYNTACTIC PATTERNS

Lexico-Syntactic Patterns (LSPs) were first applied by Hearst (1992) in the context of relation discovery from machine readable dictionaries. Hearst's patterns had prepositional phrases, paralinguistic signs or conjunctions as main elements. The object of her research was to identify hyperonymy-hyponymy relations between concepts. Since then, many authors have followed Hearst's approach for the automatic discovery of semantically related lexical items from unstructured texts with different purposes. In Ontology Engineering, LSPs have been mainly applied in two directions: 1) to learn classes, attributes or instances for ontology population (Aussenac-Gilles and Jacques, 2006; Berland and Charniak, 1999; Cimiano and Wenderoth, 2007; Etzioni *et al.*, 2004; Pasca, 2005; Reinberger and Spyns, 2004; among others), and 2) to learn taxonomic and non-taxonomic relations for ontology building (Kavalec and Svátek, 2005; Sánchez and Moreno, 2008; among others).

For the purposes of this research, we will focus on two specific sets of patterns that represent two of the most important taxonomical relations in ontologies: the *SubclassOf* relation and the *Part-Whole* relation. As already mentioned in section 2, in the two activities that concern us here, namely, ODPs reuse and ontology statement validation, these two relations are the source of many modelling errors. For that reason, we will concentrate when showing how LSPs can contribute to solve novice users' problems in reusing the ODPs to model those relations. Besides that, LSPs will be applied to discern if an ontology statement models correctly a SubclassOf or a Part-Whole relation. We will contribute to the LSPs repository in (Aguado de Cea *et al.*, 2008) by proposing some enhancements to the LSPs present in the SubclassOf and Part-Whole relations.

It is unusual to find a one-to-one correspondence between LSPs and ODPs. ODPs are classified according to the modelling problem they address, but in NL different modelling aspects can be linguistically realized with one and the same lexical element. In this case we have a NL statement corresponding to a *combination of ODPs*. For example, the verb *to classify in/into* implies the arrangement of objects in different categories that do not normally share instances, i.e., it combines the SubclassOf relation ODP with the Disjoint Classes ODP. Apart from that, sometimes the same linguistic construct can convey two different relations, or, what is the same, correspond to two or more *pairwise disjoint ODPs*. An example of this is the verb *to include*, which can either list the types into which a certain class is divided, or the parts into which an object is divided (this can also be applied to the verb *to divide*). In the latter case, a further disambiguation process is required. Let us illustrate these cases with the LSPs in the tables (The set of restricted symbols and abbreviations used is included in Appendix):

Table 1: LSPs corresponding to SubclassOf or Part-Whole ODPs

	LSP Identifier : LSP-SC-PW-EN
	NeOn ODPs Identifier : LP-SC-01 / CP-PW-01
	Formalization
1	NP<class> include | comprise | consist of [(NP<class >,)* and] NP<class>
2	NP<class> be divide | split | separate in|into [CD] [CN] [(NP<class >,)* and] NP<class>
	Examples
1	*(a) Common mass storage devices include disk drives and tape drives.* (SC) *(b) Reproductive structures in female insects include ovaries, bursa copulatrix and uterus.* (PW)
2	*(a) Marine mammals are divided into three orders: Carnivora, Sirenia and Cetacea.* (SC). *(b) The cerebrum is divided into two major parts: the right cerebral hemisphere and left cerebral hemisphere.* (PW)

In Table 1 we present the LSPs for the *pairwise disjoint* ODPs SubclassOf and Part-Whole, signalized by the use of "/" in the *NeOn ODPs Identifier* (taken from the ODPs repository in (Suárez-Figueroa *et al.*, 2007)). In this *Identifier*, LP stands for Logical Pattern and CP for Content Pattern. The *LSP Identifier* in the LSPs template consists of an acronym of the relation captured by the ODP (SC for SubclassOf and PW for Part-Whole), and the ISO-639 code for the language to which the LSP belongs.

As already mentioned, patterns in Table 1 contain polysemous verbs conveying both ontological relations, SubclassOf and Part-Whole. We have contributed to the LSPs repository by including a new verbal form in LSP 1, namely, the verb *to consist of*.

In Table 2, LSPs corresponding to the SubclassOf relation are shown. In this paper, some

modifications to the LSP-SC-EN in Aguado de Cea *et al.* (2008) have also been made, namely:

Table 2: LSPs corresponding to SubclassOf relation ODP

	LSP Identifier: LSP-SC-EN
	NeOn ODPs Identifier : LP-SC-01
	Formalization
1	[(NP<subclass>,)* and] NP<subclass> be [CN] NP<superclass>
2	[(NP<subclass>,)* and] NP<subclass> (classify as) | (group in|into|as) | (fall into) | (belong to) [CN] NP<superclass>
3	There are CD | QUAN [CN] NP<superclass> PARA [(NP<subclass>,)* and] NP<subclass>
4	[A(n) | QUAN] example | examples | [CN] of NP<superclass> be | include [(NP<subclass>,)* and] NP<subclass>
	Examples
1	*Odometry, speedometry and GPS are types of sensors.*
2	*15,5% of Spaniards are classified as obese.* *Thyroid medicines belong to the general group of hormone medicines.*
4	*There are several kinds of memory: fast, expensive, short term memory, and long-term memory.*
5	*Some examples of peripherals are keyboards, mice, monitors, printers, scanners, disk and tape drives, microphones, speakers, joysticks, plotters and cameras.*

a) LSPs 1 and 2 in the original repository have been merged into LSP 1 here;

b) A new verbal form has been included in the new LSP 2, namely, *to classify as*, and Class Name (CN) has been turned into an optional element;

c) LSP 4 in the original repository has proven to belong to the LSP for "SubclassOf relation, Disjoint Classes, and Exhaustive Classes ODPs" (see pattern number 3 in Table 3) and has been moved to that LSP. This decision is supported by a preliminary experiment reported in (Aguado de Cea and Montiel-Ponsoda, 2009) that has demonstrated that verbs such as *to classify in/into, to divide in/into, to split in/into, and to separate in/into* predicate about a set of subclasses that do not share any individuals (disjointness). Although more evidence is required, we would dare to say that for the rest of patterns represented in Table 2, no assertions can be made about disjointness or exhaustiveness without further evidence;

d) A new LSP has been included in LSP-SC-EN, namely, LSP 4.

Regarding the LSPs corresponding to "SubclassOf relation, Disjoint Classes, and Exhaustive Classes ODPs" (see Table 3), a new pattern has been included, namely, LSP 1. In this case, one LSP corresponds to a combination of three ODPs, and it is signalized by the use of "+" in the *NeOn ODPs Identifier*. Moreover, we claim that exhaustiveness is also a feature of the linguistic constructs identified in this case, although further evidence is needed.

Table 3: LSPs corresponding to SubclassOf relation, Disjoint Classes, and Exhaustive Classes ODPs

	LSP Identifier : LSP-SC-Di-EC-EN
	NeOn ODPs Identifier: LP-SC-01 + LP-Di-01 + LP-EC-01
	Formalization
1	NP<superclass> be | CATV [either] NP<subclass> or | and NP<subclass>
2	NP<superclass> be divide | split | separate | group in|into [either] [(NP<subclass >,)* and] NP<subclass>
3	NP<superclass> CATV [CD] [CN] [PARA] (NP<subclass>,)*and NP <subclass>
	Examples
1	*Animals are either vertebrates or invertebrates.*
2	*Sensors are divided into two groups: contact and non-contact sensors.*
3	*Membrane proteins are classified into two major categories, integral proteins and peripheral proteins.*

Table 4: LSPs corresponding to Part-Whole relation ODP

	LSP Identifier: LSP-PW-EN
	NeOn ODPs Identifier: CP-PW-01
	Formalization
1	(NP<part>,)* and NP<part> COMP [CN] NP<whole>
2	NP<whole> be COMP [CN] (NP<part>,)* and NP<part>
3	The part | parts of a NP<whole> be [PARA] NP<part>,)* and NP<part>
	Examples
1	*Proteins form part of the cell membrane.*
2	*Water is made up of hydrogen and oxygen.*
3	*The parts of a tree are the root, trunk(s), branches, twigs and leaves.*

In Table 4 we include those lexico-syntactic constructs that can identify a Part-Whole relation. The following modifications have been carried out to the LSP-PW-EN:

a) Verbal forms composing the set of the so-called Verbs of Composition (COMP), to which the verbs *to contain* and *to hold* have been added, and the verb *to consist of* has been removed;

b) LSP 3 has been included.

Finally, in Table 5 we show the *pairwise disjoint* ODP LSP-OP-DP-PW-EN from Aguado de Cea *et al.* (2008) that contains a polysemous verb (*to have*), which can also correspond to different modelling

patterns. In this case, disambiguation is also related with modelling decisions, i.e., if we have already discerned that we are not dealing with parts of an object (PW), but with properties, we have to decide if we want to model a certain property as an ObjectProperty (OP), or as a DatatypeProperty (DP).

Table 5: LSP corresponding to Object Property, Datatype Property and Part-Whole relation ODPs

LSP Identifier : LSP-OP-DP -PW-EN	
NeOn ODPs Identifier: LP-OP-01 / LP-DP-01 / CP-PW-01	
Formalization	
1	NP<class> have NP<class>
Examples	
1	*Birds have feathers.* *Water areas have names in natural language.*

4 METHOD FOR ODP REUSE

The repository introduced in section 3 is the core element of the method we present here. The main purpose of this repository is to bridge the gap between NL and the modelling solutions represented by ODPs. In this way, this method aims at enabling an easier reuse of ODPs to users with different expertise in Ontology Engineering. The whole process needs to be supported by a system that automatically analyses the sentence in NL introduced by the user, and looks for the linguistic structure that best matches the input in the LSPs repository. However, some LSPs are ambiguous, i.e. the same linguistic structure can equally express the information encoded in *pairwise disjoint* ODPs (as already exemplified in Tables 1 and 5). This means that the user input may need to be revised or refined during the process, so that only one option is valid. The method is mainly divided in three tasks:

Task 1. Formulation. The goal of this task is to formulate in NL the domain aspect to be modelled. We assume that the user has a good command of the knowledge parcel (s)he wants to model in the ontology. Contrary to the proposal made in (Aguado de Cea *et al.*, 2008), in which the user was allowed to introduce in *full NL* the domain aspect to be modelled, we claim here that some recommendations on the use of NL should be made in order to guarantee the results. For example, to express sentences in present tense; to express one topic or idea per sentence; to avoid coordination of phrases, that will be only used when necessary; to avoid redundant information (adjectives, adverbs) or modal verbs; or to use full stop to end up sentences, among others.

Task 2. Refinement. The goal of this task is to refine the input because an LSP matches *pairwise disjoint* ODPs. This task will take place when LSPs of the type represented in Table 1 and Table 5 are matched. Different strategies for the performance of this task are being investigated, such as interaction with the user; search in available ontologies modelling the same domain of knowledge; search in lexical resources with some semantics (e.g. WordNet); etc.

Task 3. Validation. The goal of this task is to confirm that the ODP proposed as modelling solution is correct. The validation is foreseen to be manually carried out by the user.

4.1 Example of use

In this section, our aim is to exemplify the proposed method for the reuse of ODPs having as starting point sentences in NL. Since the relations dealt with here are SubclassOf and Part-Whole, we will focus on the employment of the proposed method with one example of a polysemous LSP conveying both ontological relations (see Table 1). The method could help in the following way:

Task 1. Formulation. Let us assume that the user wants to model the Part-Whole relation holding between the brain (cerebrum) and its parts (the right hemisphere and the left hemisphere). The first task consists in formulating that knowledge according to the recommendations introduced in the method. If we take Example 2 (b) in Table 1: *The cerebrum is divided into two major parts: the right cerebral hemisphere and left cerebral hemisphere*, the adjective "major" appearing in this sentence should be removed, since it is redundant here.

Task 2. Refinement. The sentence in NL matches the LSP-SC-PW-EN corresponding to *pairwise disjoint* ODPs: LP-SC-01 and CP-PW-01 (see Table 1). There would be two options for refining the input: 1) interaction with the user by asking him or her if the *right cerebral hemisphere and left cerebral hemisphere* are "types of" cerebrum or "parts of" cerebrum; 2) search for *cerebral hemisphere* in a lexical or terminological resource such as WordNet and check which relation it bears to cerebrum.

Task 3. Validation. In return, the user receives an ODP represented by a UML diagram instantiated with the information of the input sentence in Task 1. Then, the user has to confirm the ODPs suitability.

5 VALIDATION OF ONTOLOGY STATEMENTS

The repository of LSPs introduced in section 3 can also be very useful when verifying the quality of the ontology statements the user wants to reuse for ontology building or enrichment. Our starting hypothesis here was that if the knowledge expressed in the ontology statement was formally correct from a modelling perspective, it had to be found in domain texts expressed by means of the linguistic constructs identified as usually conveying the relation represented in the statement.

This means that we could expect to find the concepts in the statement (let us say, *Branch part_of Tree*), forming part of NL structures identified in the LSPs repository as conveying the Part-Whole relation (e.g. *Trees have branches*). On the contrary, if the relation said to be hold between the ontology concepts in the statement were (formally) false (*Branch subclass_of Tree*), we would not find sentences expressing SubclassOf relation (*Trees are classified into branches* or *Branches are trees)*, or at least a smaller quantity. The proposed approach to validate ontology statements before reuse can be summarized in three steps:

Step 1. LSPs Instantiation. The purpose is to instantiate the set of LSPs in the repository with the ontology concepts in the statement to be reused. By instantiation we understand the replacement of Noun Phrases (NP) contained in LSPs, with the ontology labels in ontology statements.

The following aspects need to be taken into account before proceeding with the instantiation:

1. Ontology labels have to be used in its plural form. In ontologies, labelling style guides (see Fliedl *et al.*, 2007) recommend to express labels that designate concepts in singular. However, when referring to classes in general, the English language makes use of the plural form of nouns. For instance, *tress have branches*. Therefore, the first thing is to obtain the plural form of each ontology label.
2. Optional elements have to be removed, thus maintaining the basic form of the pattern. The set of optional elements that appear in LSPs are included in square brackets, and, if omitted, they do not modify the basic meaning conveyed by the pattern. The reason for using the pattern in its basic form is motivated by the effort required by the manual instantiation of LSPs with optional elements. Should the approach be automated, this problem would disappear.
3. The semantic role of the concepts in the ontology statement is maintained in the pattern instantiation. Semantic roles are signalized in LSPs by means of angle brackets (<superclass>), as detailed in the Appendix at the end of the paper. Semantic roles in LSPs are closely related with the grammatical function of a certain Noun Phrase in the pattern, and are essential components of LSPs. As a first approach to the modelling error presented here, we create equivalences between "superclasses" and "wholes", and "subclasses" and "parts", and keep those roles in the instantiation. However, this strategy is just one of several possible strategies. We are now investigating the possibility of instantiating LSPs allowing the two labels in the ontology statements to play any semantic role in LSPs.

Step 2. Web Search. The purpose is to introduce the resulting NL sentences in a Web search engine to check their frequency in domain texts.

Step 3. Validation. The purpose of this step is to evaluate the resulting matches for each LSP. The instantiations of those LSPs conveying the relation that really holds between the ontology concepts should appear in Web documents more frequently.

In the near future, the idea would be to automate these steps to achieve an automatic validation of ontology statement before its reuse. However, up to now, we have only evaluated the proposed approach manually, as explained in section 5.1.

5.1 Example of use

In this section, our aim is to describe how to proceed with a manual validation of ontology statements with LSPs. We will concentrate on those ontology statements that incorrectly express the SubclassOf and the Part-Whole relations, since these relations account for some of the most common errors in ontologies, as already outlined in section 2.

Let us take as example the erroneous ontology statement *chapter_subclass_of_book*. This is a real example of an ontology statement accessed through Watson from a subset of the AKT Reference Ontology (Portal Ontology) written in OWL LITE (http://www.csd.abdn.ac.uk/~cmckenzi/playpen/rdf/akt_ontology_LITE.owl).

Step 1. LSPs Instantiation. The terms or labels that name the concepts "chapter" and "book" will now be used to instantiate the different LSPs in which the SubclassOf relation and the Part-Whole relation are involved, i.e., the LSPs described in section 3. Let us illustrate the instantiation process with LSP 1 in Table 2 corresponding to the SubclassOf relation:

[(NP<subclass>,) and] NP<subclass> be [CN] NP<superclass>*

1. The first action is to obtain the plural forms of the terms in question: *chapters* and *books*.
2. Optional elements are removed. This is the case of [CN] in the pattern taken as example. CN stands for "Class Name: Generic names for semantic roles usually accompanied by a preposition, such as (sub)class, type, or category" (as detailed in the Appendix). By keeping this optional element, the LSP in our example could have the following instantiations: *chapters are books*, *chapters are types of books*, *chapters are subclasses of books*, etc.
3. Semantic roles are maintained. In the present case, *chapter_subclass_of_book*, "chapter" plays the role of subclass, and "book" the role of superclass. These roles should be maintained in the instantiation of LSPs for the SubclassOf relation. In the LSPs for the Part-Whole relation, "chapter" would be considered playing the role of "part", and "book" the role of "whole".

Step 2. Web Search. Once we have instantiated all LSPs in which the SubclassOf relation and Part-Whole relation are involved (see Table 6), we proceed to manually include the resulting NL sentences into the Google search engine. The Google configuration options used in this step were:

- It should only find those pages that contained the exact wording or phrase.
- It should only look in pages in English.
- It could access files in any format.

The first configuration option is the most important, because in this way Google only counts the number of appearances in which web documents exactly contain the resulting NL sentences.

This step would result in numerical data about how many times sentences appear in Google. It is worth mentioning that Google does not give information about the total amount of indexed pages it accesses during the search (e.g., for the instantiation "chapters *are* books" Google returned 7 matches, whereas for "books *contain* chapters", it returned 547 matches).

Step 3. Validation. According to the results, the relation to be hold between the ontology concepts in the ontology statements is confirmed to be valid, or proved to be false. We are currently working on the establishment of statistical parameters to automatically validate ontology statements.

Table 6: Ontology statement instantiation example

LSP-SC-EN / LSP-SC-Di-EC-EN	
1	"chapters *are* books"
2	"chapters *are grouped into* books"
3	"chapters *fall into* books"
4	"chapters *belong to* books"
5	"*examples of* books *are* chapters"
6	"books *are classified into* chapters"
LSP-SC-PW-EN	
7	"books *include* chapters"
8	"books *consist of* chapters"
9	"books *are divided into* chapters"
LSP-PW-EN / LSP-OD-DP-PW-EN	
10	"chapters *are contained in* books"
11	"chapters *compose* books"
12	"chapters *make up* books"
13	"chapters *are part of* books"
14	"books *are made up of* chapters"
15	"books *contain* chapters"
16	"books *have* chapters"

6 CONCLUSIONS

The main purpose of the work presented here was to show the suitability of Lexico-Syntactic patterns (LSPs) in the development of ontologies, particularly, in the reuse of Ontology Design Patterns (ODPs), and for the validation of ontology statements. The initial hypothesis was that some linguistic constructs reliably convey ontological relations. By establishing correspondences between LSPs and ODPs in the LSPs repository, we provide a method that enables novice users the formulation of modelling aspects in NL, which are then transformed into ODPs. Additionally, the LSPs repository can be employed in the quality assessment of ontology statements, since the information captured in the statement has its correspondence to expressions in NL collected in the repository. Our proposal has been detailed and illustrated in two examples of use. The examples already point out some of the future lines of research, such as the establishment of statistical parameters for analyzing "ontology statements validation results" (Step 3). Additionally, experiments are also needed to test the proposed method for the reuse of ODPs, and to further improve the LSPs repository.

REFERENCES

Aguado de Cea, G., and Montiel-Ponsoda, E. 2009. When Phraseology and Ontologies Meet. In Proccedings of

the *XI Simposio Internacional de Comunicación Social*. Santiago de Cuba, Cuba.

Aguado De Cea, G., Gómez-Pérez, A., Montiel-Ponsoda, E., and Suárez-Figueroa, M.C. 2008. Natural Language-Based Approach for Helping in the Reuse of Ontology Design Patterns. In *Proceedings of the 16th International Conference on Knowledge Engineering (EKAW)*, pp. 32-47.

Aussenac-Gilles, N., and Jacques, M.P. 2006. Designing and Evaluating Patterns for Ontology Enrichment from Texts. In Staab, S. and Svatek, V. (ed.) *Managing Knowledge in a World of Networks, 15th International EKAW Conference*, Springer Verlag.

Berland, M., and Charniak, E. 1999. Finding parts in very large corpora. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, College Park, MD.

Cimiano, P. 2006. Ontology Learning and Population from Text. Algorithms, Evaluation and Applications. Springer Verlag.

Cimaino, P. and Wenderoth, J. 2007. Automatic Acquisition of Ranked Qualia Structures from the Web**.** In *Proc of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 888-895.

Dzbor, M., Suárez-Figueroa, M.C., Blomqvist, E., Lewen, H., Espinoza, M., Gómez-Pérez, A., and Palma, R. 2009. NeOn Deliverable D5.6.2 Experimentation and Evaluation of the NeOn Methodology.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D., and Yates, A. 2004. Methods for domain-independent information extraction from the web: An experimental comparison. In *Proceedings of the 19th National Conference on Artificial Intelligence (AAAI)*, pp. 391-398.

Fliedl, G., Kop Ch., and Vöhringer, J. 2007. From OWL Class and Property Labels to Human Understandable Natural Language. In *Natural Language Processing and Information Systems*, Volume 4592/2007, Springer Berlin / Heidelberg.

Gangemi, A. 2005. Ontology Design Patterns for Semantic Web Content. In: Musen et al. (eds.): *Proc. of the 4th ISWC*, Springer, Heidelber.

Hearst, M. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING)*, pp. 539-545.

Kavalec, M. and Svátek, V.2005. A study on automated relation labelling in ontology learning. In Buitelaar, P., Cimiano, P., Magnini, B.(eds.) *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press, pp. 44-58.

Menzies, T. 1997. Object-oriented patterns: Lessons from expert systems. *Software-Practice and Experience, 1(1)*.

Pasca, M. 2005. Finding Instance Names and Alternative Glosses on the Web: WordNet Reloaded. In *Computational Linguistics and Intelligent Text Processing*, Springer Berlin/Heidelberg, pp. 280-292.

Reinberger, M.L., and Spyns, P. 2004. Discovering knowledge in texts for the learning of dogma-inspired ontologies. In *Proceedings of the ECAI Workshop on Ontology Learning and Population*, pp. 19-24.

Sabou, M., Aguado de Cea, G., d'Aquin, M., Daga, E., Lewen, H., Montiel-Ponsoda, E., Presutti, V. and Suárez-Figueroa, M.C. 2009. NeOn D2.2.3. Evaluation and Selection of Knowledge Components.

Sabou, M., d'Aquin, M., and Motta, E. 2008. Exploring the Semantic Web as Background Knowledge for Ontology Matching. *Journal on Data Semantics*, XI.

Sánchez, D., and Moreno, A. 2008. Learning non-taxonomic relationships from web documents for domain ontology construction. In *Data Knowledge Engineering, 64*, pp. 600-623.

Suárez-Figueroa, M.C., Brockmans, S., Gangemi, A., Gómez-Pérez, A., Lehmann, J., Lewen, H., Presutti, V. and Sabou, M. 2007. NeOn D5.1.1. NeOn Modelling Components.

Suárez-Figueroa, M.C. *et al.* 2008. NeOn D5.4.1. NeOn Methodology for Building Contextualized Ontology Networks.

AKNOWLEDGEMENTS

This research has been supported by the European project NeOn (FP6-027595), and the National project GeoBuddies (TSI2007-65677C02).

APPENDIX

SYMBOLS & ABBREVIATIONS in LSPs	
CATV	Verbs of Classification. Set of verbs of classification plus the preposition that normally follows them. Some of the most representative verbs in this group are: *classify in/into, categorize in/into, subclassify in/into, subcategorize in/into*.
CD	Cardinal Number.
CN	Class Name. Generic names for semantic roles usually accompanied by preposition, such as *class, type, category*
COMP	Verbs of Composition. Set of verbs meaning that something is made up of different parts. Some of the most representative ones are: *contain, hold, consist of, compose of, make up of, form of/by, constitute of/by*.
NP<…>	Noun Phrase. It is defined as a phrase whose head is a noun or a pronoun, optionally accompanied by a set of modifiers, and that functions as the subject or object of a verb. NP is followed by the semantic role played by the concept it represents in the conceptual relation in question in <…>, e.g., *class, subclass*.
PARA	Paralinguistic symbols like *colon*, or more complex structures as *as follows*, etc.
QUAN	Quantifiers such as *all, some, most, many, several, every*.
( )	Parentheses group two or more elements.
*****	Asterisk indicates repetition.
[ ]	Elements in brackets are meant to be optional, which means that they can be present either at that stage of the sentence or not, and by default of appearance, the pattern remains unmodified.