# Promoting Scientific Creativity by Utilising Web-based Research Objects

Integrated Project (IP)

FP7-ICT-2013-10. Information and Communication Technologies

Grant Agreement Number 611383

José Luis Redondo García, David Chaves Fraga, Carlos Badenes Olmedo, Óscar Corcho

**Universidad Politécnica de Madrid**

**Deliverable D5.5**

## Final Version of Ontology Learning and Matching Techniques with Report

# Grant agreement no: 611383

| Dissemination Level | | |
|---|---|---|
| **PU** | Public | |
| **PP** | Restricted to other programme participants (including the Commission Services) | |
| **RE** | Restricted to a group specified by the consortium (including the Commission Services) | **X** |
| **CO** | Confidential, only for members of the consortium (including the Commission Services | |

| *COVER AND CONTROL PAGE OF DOCUMENT* | |
|---|---|
| Project Acronym: | Dr Inventor |
| Project Full,Name: | Promoting Scientific Creativity by Utilising Web-based Research Objects |
| Deliverable No.: | D5.5 |
| Document name: | Final Version of Ontology Learning and Matching Techniques with Report |
| Nature (R, P, D, O): [1] | Report |
| Dissemination Level (PU, PP, RE, CO): [2] | RE |
| Version: | v1.0 |
| Actual Submission,Date: | 21/11/2016 |
| Internal Reviewer: | Horacio Saggion |
| Editor:<br>Institution:<br>E-Mail: | José Luis Redondo García<br>Universidad Politécnica de Madrid (UPM), Madrid, Spain<br>jlredondo@fi.upm.es |

**ABSTRACT:**

Ontology Learning algorithms are used to automatically generate ontologies, usually from unstructured resources. They are specially useful in particular domains where there are no mature or de-facto ontologies available. As part of the DRInventor research efforts, we have developed an ontology learning algorithm that aims to generate the domain model underlying the research objects indexed in the platform. Our approach is able to detect a set of relevant terms and relations that annotate the information contained in the corpus's resources. We also present a unified framework for evaluating ontology learning algorithms, which takes into consideration different lexical and taxonomical aspects that are compared against a semi-automatically generated Gold Standard.

[1] **R**=Report, **P**=Prototype, **D**=Demonstrator, **O**=Other

[2] **PU**=Public, **PP**=Restricted to other programme participants (including the Commission Services), **RE**=Restricted to a group specified by the consortium (including the Commission Services), **CO**=Confidential, only for members of the consortium (including the Commission Services)

**KEYWORDS LIST:**
Ontology Learning, Evaluation, Term Extraction, Gold Standard.

*The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7-ICT-2013.8.1) under grant agreement no 611383.*



MODIFICATION CONTROL

| Version | Date | Status | Author | Comments |
|---|---|---|---|---|
| 0.1 | 02/09/2016 | Draft | José Luis Redondo García | Initial version of 5.5 |
| 0.2 | 20/09/2016 | Draft | José Luis Redondo García | Adding O.Learning Experiments and Eval |
| 0.3 | 02/10/2016 | Draft | David Chaves Fraga | Adding comments on evaluation settings |
| 0.4 | 04/10/2016 | Draft | José Luis Redondo García | Reviewing David's suggestions |
| 0.5 | 08/10/2016 | Draft | Óscar Corcho | Rephrasing and quality check |
| 0.6 | 12/10/2016 | Draft | José Luis Redondo García | Reviewing Oscar's suggestions |
| 0.7 | 03/11/2016 | Draft | José Luis Redondo García | Applying Internal Review comments |
| 1.0 | 15/11/2016 | Final | José Luis Redondo García | Minor changes and generating Final version |

## Executive Summary

*Ontology Learning* algorithms are used to automatically generate ontologies, usually from unstructured resources. They are specially useful in particular domains where there are no mature or de-facto ontologies. This discipline is re-flourishing thanks to recent advances in information extraction techniques exploiting novel features and paradigms such as Word2Vec and Deep Learning.

In this deliverable we present the ontology learning strategy developed under the DRInventor project, that has been applied over the corpora of Research Objects indexed by the platform. The reasons for leveraging on such techniques for annotating the DRInventor corpus are mainly two: 1) getting insights about the kind of knowledge represented in the corpus through the most relevant concepts found in the resources, and 2) empower advanced operations that can benefit from a better contextualisation that relevant terms and relations between terms can provide, such as recommending pertinent research objects to users accessing the platform.

The implemented ontology learning approach is able to spot two kind of elements: relevant **terms** that represent the most important concepts describing the domain, and **relations** between those terms in order to discover how they are semantically connected and get better insights about how the knowledge inside the corpus is organised. The main innovation aspects we have focused on when developing such algorithms are:

1. In the case of the terms, we have made special emphasis on tailoring as much as possible the results of ontology learning algorithm to the domain represented in the corpus. In a scientific context, this is a very important aspect to be taken into account if we want to obtain relevant results for the research task being addressed. We have combined different techniques [31] [15] aiming to determine the *term-hood* of the words found in a corpus, in order to better judge on the importance of those terms inside the entire knowledge contained in the research objects.

2. For the relations or connections between terms, we have dealt with the complexity of applying supervised techniques for solving such kind of complex problems where even human experts struggle to provide results. In most cases the training phase of those approaches requires a huge amount of data that is not usually available due to this difficulty in manually spotting them. In order to alleviate the lack of labelled information about terms' connections we have leveraged on Distant Supervision [25] techniques.

Stabilising a function able to determine the success of an ontology learning approach is a very ambitious objective given the fact it usually involves many considerations and requires a strong background knowledge in order to do fair judgements over candidate terms and relations. However, the evaluation of results is a crucial step in the learning process: it allows

comparing different learning approaches in a systematic way, so that we can select the one that fits better our objectives, and it gives support to supervised approaches relying on iterative searches in the space of solutions, which need functions to determine how close they get from the optimal result.

### Summary of Novelty

State-of-the-art ontology learning approaches normally work over predefined, static corpus by trying to align concepts found in the analysed documents to already existing ontologies in the domain. In this deliverable we present a novel ontology learning method that can operate over more heterogeneous, continuously evolving data sources where no previous de-facto ontologies are available.

In addition, we present an innovative approach for evaluating ontology learning algorithms, which takes into consideration different lexical and taxonomical aspects that are compared against a semi-automatically generated Gold Standard. The designed methodology intends to minimise human intervention and promotes more replicable experimental setups. Through various use cases including the evaluation of the DRInventor corpus, we have proven how the proposed methodology can adapt to different scenario requirements while increasing the representativeness of the Gold Standard, therefore providing more meaningful insights about the candidate techniques.

# Table of Content

## 0 Deliverable Structure

The remainder of this document is organised as follows: in Section 1 we introduce the techniques described in this deliverable and justify their importance inside the scope of DRInventor Platform. In Section 2 we review the most prominent initiatives in the literature of ontology learning, and highlight some of the evaluation efforts that they have triggered.

In Section 3 we describe the ontology learning approach developed for the DRInventor Platform, focusing in how to spot both the terms and the relations that are implicitly available in the set of research objects indexed in the platform. In order to achieve this objective, various techniques from the information extraction and machine learning field are combined and applied.

A methodology for evaluating ontology learning approaches is presented in Section 4. We formalise the different considerations that should be taken into account in order systematically evaluate the generated vocabularies. Also, different use cases in the literature are presented to analyse how they can be supported by the proposed methodology. Finally, we evaluate the results obtained by our ontology learning algorithm when applied over the DRInventor corpus.

Section 5 wraps up the main contributions of this deliverable and gives a final overview about the possibilities offered by DRInventor for the learning of ontology features and its evaluation. We also sketch future lines of work for emphasising the potential of the platform for improving the way the main knowledge in the corpora is automatically extracted.

# 1 Introduction

Every second, huge amounts of data about any imaginable topic are being generated or captured, and therefore become ready to be exploited. A significant subset of this information is available in the form of natural language text, which needs to be interpreted. Unfortunately, the most advanced agents to perform such a task (we, humans) are not capable of processing a significant fraction of this data in a reasonable amount of time. In the DRInventor project, this issue is highly relevant and has been addressed: the amount of scientific papers being published every day about any imaginable research field is huge, and scientists need to be provided with mechanisms that allow them to browse pertinent information in a timely manner.

*Ontology Learning* algorithms are used to automatically or semiautomatically generate ontologies from a set of documents where the knowledge is contained, but not explicitly identified: collection of textual articles, tables in databases, graphs, etc. Ontologies allow users to understand, from a single information unit, the kind of knowledge being represented by the underlying documents. They also help machines to better exploit data by contextualising and customising different operations over the items inside the corpus, therefore being very attractive for DRInventor and the knowledge discovering capabilities that it implements.

Today's information extraction and knowledge representation scenarios are changing. Innovative techniques to generate features from text, such as Word2Vec, or paradigms like Big Data or Deep Learning allow processing huge amounts of information with higher accuracy and therefore target a wider variety of domains. For example the DRInventor corpus is focused around the computer graphic domain (SIGGRAPH[3] papers since 2002) and it includes recently published research objects about very innovative, yet unestablished research subdomains for which no previous knowledge is available. In this situation, the task of finding already-existing ontologies that sufficiently match the data being analysed becomes unfeasible in most of the cases. In addition, the knowledge about the domain is quickly evolving so models need to be able to capture the changeable context and react to new trends reshaping the conceptualisation over time.

Therefore, former ontology learning efforts relying on already-existing reference ontologies may not be enough for this kind of information extraction tasks. Current approaches are converging towards the need of more flexible, lightweight, local models that can closely describe the corpus and are easily updatable. As depicted on the right side of Figure 1 they are generated solely from the data with no influence of previous established models, and aiming for a less strict degree of formalisation than what traditional ontology learning approaches wanted to capture.

In the DRInventor Platform, research objects are automatically ingested and indexed according to the model described in [4]. Once they become available they are annotated using techniques such as Named Entity Extraction or Probabilistic Topic Modelling. Those resources

[3] https://www.siggraph.org/

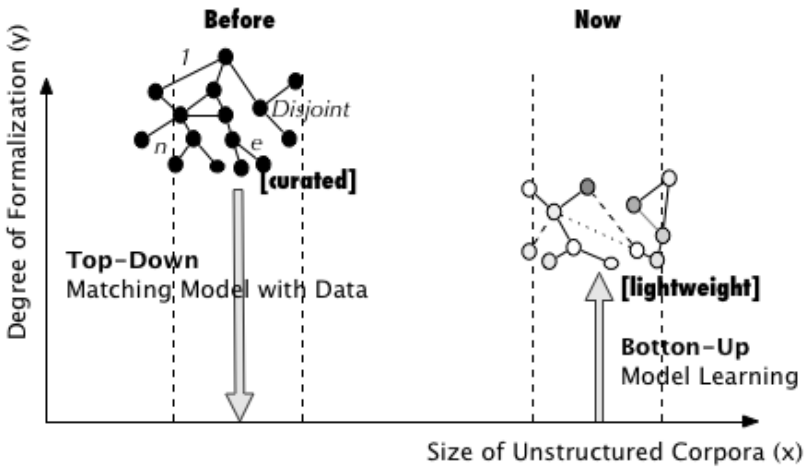


Figure 1: Increasing importance of ontology learning evaluation in current scenarios

and annotations are the starting point for a fully automatic generation of a model that tries to represent the knowledge available in the corpora. We will develop an approach that performs this ontology creation process by spotting relevant terms and relations, this way they become available to support advanced browsing and recommendation operations like the ones described in [29]. The exploitation of those new annotations and features is highly relevant and yet to be explored.

In order to check on the quality of those automatically generated models, in this deliverable we will also propose a new evaluation framework to deal with the new requirements of the reemergent field of ontology learning, aiming to identify an incremental set of evaluation objetives and methods that can lead to more effective judgements about the quality of the learned model.

The same way the approaches are evolving, the methods for evaluating them have to adapt accordingly. For example, former frameworks in the research field such as [9] tackled this evaluation by comparing the generated vocabularies against some well-stablished ontologies in the domain. Consequently, the objective was to build a model with the same level of generalisation and high formal restrictions than the reference ontology, which has been normally engineered by humans. Given that a domain ontology is already available, the learning process becomes less important and approaches tend to adopt a top-down paradigm (see left side of Figure 1) where data is matched to the model and not the other way around [2]. Nowadays, the evaluation methods are targeting lighter models (right side of Figure 1) that have been automatically inferred from the corpus, in order to check more on the adequacy of terms and certain relations, and less on the formal ontological aspects that are not so relevant for the task considered.

## 2 State of the Art in Ontology Learning

Ontology Learning is a wide discipline that considers a great variety of methods and techniques, some of them reported in [5]. In the particular case of ontology learning from text, which is the main focus of our work, some relevant approaches have been described in [36] and also in [19]. Probably the most significant examples of systems performing Ontology Learning are OntoLearn [35] and CRCTOL [22]. The former leverages on some information extraction techniques to identify different aspects from the original corpus, resulting a very dense, cyclic and potentially disconnected hypernym graph. The algorithm then induces a taxonomy from this graph via optimal branching and a weighting policy. The latter (CRCTOL) employs a combination of statistical and lexico-syntactic methods, including a statistical algorithm that extracts key concepts from a document collection, a word sense disambiguation algorithm that turns words in into dereferencable concepts, a rule-based algorithm that extracts relations between the key concepts, and a modified generalised association rule mining algorithm that prunes unimportant relations for ontology learning.

Below we enumerate other systems implementing different ontology learning methods. The system Hasti [32] builds ontologies in an incremental way, starting from natural language texts where the new relevant words that are found are added to a lexicon. It starts by generating a kernel of just a few high ranked concepts, and continues by adding other important terms and relations following an hybrid symbolic approach combining linguistic, logical, template driven and semantic analysis methods. Syndikate [21] is a system for automatically acquiring knowledge from real world. It analyses single sentences, but it also considers concepts potentially linking those sentences and forming cohesive texts. In order to decide on the relevance of new concepts, Syndikate relies on the prior knowledge of the domain the texts are about, and the gramatical structures in which potential terms can be acquired from the corpus. Text2Onto [9] is an ontology learning environment providing a general architecture for discovering conceptual structures and ontologies from text. In addition it supports the mapping of external linguistic resources to the acquired structures. The new version is focused on learning ontologies from Web documents by implementing different methods for importing semi structured and structured data such as tables inside the documents. Some of the functionalities are exposed as a library of learning methods which can be used demand. Other systems have tackled the ontology learning process by considering a certain amount of human intervention. For example Ontogen [14] framework integrates machine learning and text mining algorithms into an efficient user interface, lowering the entry barrier for users who are not professional ontology engineers. The main features of the systems include unsupervised and supervised methods for concept suggestion and concept naming, as well as ontology and concept visualisation. Also, other systems such as Welkin [3] do not aim at completely building ontologies from the scratch. Instead, they try to automatically enrich existing general purpose ontologies in order to extend them or better contextualising them.

The ontology learning approach presented in this deliverable is highly inspired in the following research works:

- For the generation of terms we leverage on approaches calculating the term-hood of a word (the relevance according to the corpus where this word has been taken from). In [31] and in [15] the authors apply different statistical methods for inferring the likelihood of a term to be significant inside the knowledge of a given corpus. Normally, frequency based an entropy-related measures are computed in order to be able to judge on the importance of the studied term. In our approach we will reuse some of those techniques by combining them into a single score that allows distinguishing between the meaningful concepts and the ones that are not representative for the domain and therefore can be discarded.
- For the generation of relationships between terms, our systems relies on a variant of Distant Supervision, as described in [25]. In particular, we use a general database for performing the training, and we rely on different relational and syntactical patterns inside the sentences to find appropriate features to look at.

Concerning the evaluation of ontologies, our methodology has been initially inspired by the paper by Dellschaft et al. [12], which already formalises an approach for evaluating ontology learning algorithms. The main difference with our work is that their evaluation is grounded on the existence of de-facto ontologies, so the utilised methods come down to the application of ontology alignment techniques. In similar work from the same authors [11] the corpus-based methods are described as adequate for ontology learning evaluation. We have revisited this idea for our methodology, in an attempt to make it more lightweight and in line with current requirements of the ontology learning task.

Different ontology learning systems have also presented their own methodology for evaluating the results. The ontology learning system Syndikate implemented an incremental algorithm exploiting evidences and certain credibility hypotheses about concepts, which are refined through a supervised classification method trained on related knowledge bases. The evaluation combined a comparison to an already-existing corpora from information technology with some manual assessments on the generated hypothesis, making this method hardly reproducible by others. Text2Onto [9] selected another well-known field (tourism), and compared their approach against an already available domain ontology for applying precision and recall measures on terms and relations. The set of unnamed, non-taxonomic relations were hand-coded into the very same ontology before. Hence, they incurred on significant costs derived from the human efforts made, what we try to alleviate in our proposal. The system at [30] presents an approach for learning ontologies in bioinformatics. In this case, the evaluation methods verified the quality of the generated ontologies by comparing them against reference ontologies which were not initially available, but were hand-built for the occasion.

A more recent example is the framework Galeon [24], which performs an evaluation over terms and hypotheses (relations) from a traditional ontology-matching oriented point of view, by comparing the learned ontology with reference ontologies in the domains of universities and economics. In [8] they use an interesting mechanism to build a synthetic dataset to compare with, however we argue wheter the absence of lightweight human intervention can still lead to quality Gold Standards. They also perform a criteria-based evaluation considering aspects from the graph/tree theory literature, such as “Mean to Root” or “Mean to Parent”[4]. However we will opt for more functional measures, which are easier to interpret.

Finally, CRCTOL [22] distinguishes between a “component level” and an “ontology level" during the evaluation. The former phase consists on a lexical comparison with a gold standard in the terrorism and sport domains in order to quantify the performance against other systems like Text2Onto. The latter performs an evaluation on the relations learned, using quantitative and qualitative methods, and including an analysis of the graph structural properties, comparison to WordNet, and expert rating. This is probably the most exhaustive evaluation methodology that can be found in the literature, but it is not properly formalised: the gold standard annotations were generated from scratch by humans, and human efforts are not always reproducible. From the different alternatives studied here, we can observe how none of them has followed a well formalised, easily applicable procedure sufficiently aiming to reduce the high costs derived from human intervention. Therefore in Section 4 we will propose a methodology to overcome those problems.

[4] https://en.wikipedia.org/wiki/Tree_(data_structure)

# 3 The DRInventor Ontology Learning Approach

In this section we describe the term extraction approach implemented in the DRInventor Platform. We make special emphasis in highlighting the different technical decisions taken for improving the way the most relevant terms and relations in the corpus are spotted.

First, the implemented algorithm is able to perform a domain-dependent extraction of relevant terms by combining different relevance dimensions into a single approach: domain consensus term-hood, domain relevance, and C-value/NC statistical measures that are exploited together to improve the state-of the-art approaches. Secondly, the learning algorithm is also able to find hypernymy relations between terms by performing a distantly supervised learning over a general corpus (Wikipedia).

## 3.1 Term Extraction

The most important phase inside the designed ontology learning process is terminology extraction, which consists in obtaining the most important concepts in the domain by combining methods leveraging on statistical domain relevance and pertinence indicators. The input for the terminology extraction task is the set of terms that are included in the collection of research objects indexed in the platform. The implemented terminology extraction method is divided into the following steps:

- Domain linguistic annotation. A linguistic annotation process is performed over each document belonging to the domain corpus, by invoking the NLP service[5]. Apart from the generic techniques commonly considered in most of the NLP tasks (such as automatic Part of Speech[6]), the DRInventor NLP service also performs a *term candidates extraction* process. A transducer is run over each sentence in the documents in the domain corpus in order to find candidates, which can be seen as noun phrases that structurally seem to be terms, yet they might not meet consensus and relevance requirements. The term candidates transducer executes a grammar that formalises some pre-defined noun phrases cases taken from [15], which have been identified to ideally not harm precision.

- Termhood calculation. For each term candidate detected by the transducer in the previous step the Learner calculates its term-hood, which is determined by the following measures:

  1. Domain Consensus [31]. An entropy-related measure that expresses how spread is the use of a certain term throughout the documents of the considered corpus, since this distributed usage would somehow express a form of agreement and consensus.

[5] http://backingdata.org/dri/library/
[6] https://en.wikipedia.org/wiki/Part_of_speech

It is computed as follows:

$$D_C(t) = -\sum_{d \in D} (P(t/d)) \log (P(t/d)) \tag{1}$$

Where $d$ is any document belonging to the domain $D$, and $t$ is a candidate term inside the corpus.

2. Domain Relevance [31]. A probabilistic measure of how relevant a term $t$ is in a domain $D$ with respect to other domains. It is expressed as the probability of the term appearing in the considered domain $D$ divided by the probability of such term in the domain where is it is more likely to be appearing. Formally it can be expressed as:

$$D_R(t) = \frac{P(t/D)}{\max_k P(t/D_k)} \tag{2}$$

3. C-value. This statistical measure is based on the C-value/NC-value described in [15], but leaving out the contextual variables to keep the calculation simpler.

$$C_{Value}(t) = \begin{cases} \log_2 \|t\| \cdot \#(t), \text{if } t \text{ has not superterms} \\ \log_2 \|t\| \cdot \left[ \#(t) - \frac{1}{\sup(t)} \left[ \sum_{st \in \sup(t)} \#(st) \right] \right], \quad \text{otherwise} \end{cases} \tag{3}$$

Where:

   - $\#(t)$ represents the number of occurrences of the term $t$ in the considered domain.
   - $\|t\|$ represents the length of the term $t$.
   - $\sup(t)$ is the set of superterms or terms found in the domain in which $t$ is contained.

4. The term-hood of a term candidate is finally determined as the linear combination of the previous measures (those that are not probabilistic are normalised to the $[0,1]$ range). Formally we write:

$$\text{termhood}(t) = \alpha * \text{norm}(C_{Value}(t)) + \beta * \text{norm}(D_C) + \gamma * D_R \tag{4}$$

In the current implementation of DRInventor Term Extraction modules, the three different measures have been assigned the same importance when combining them together ($\alpha = \beta = \gamma = 0.\overline{3}$). In future work we plan to optimise this combination by adapting the weights to the specific corpus analysed.

## 3.2 Relations Extraction

For the extraction of relations we propose the use of a distantly supervised approach with a schema of one relation extractor per each type of relation. Currently, only *hypernymy relations* are supported. The intuition of distant supervision (term coined by [25] but initially proposed by [33]) is that any sentence that contains a pair of entities that participate in a known and curated knowledge base relation is likely to express that relation. It can also be seen as a case of the noisy channel model, in which there is a knowledge base that acts as an oracle that expresses clearly the relationships between knowledge entities, and such information is scrambled thorough the textual domain corpus. Using a distant supervision approach we aim at learning the patterns of how this information is transformed into the sentences where these entities appear. Initially we use the intuition from [25] and assume that such relations manifest themselves in a sentence basis (we leave as future line of work the consideration of multiple sentences).

The main advantage of using a distant supervised method is that the training corpus can be separated from the underlying domain. Statistics-based methods, which we apply, need a huge training corpus, it might be possible that the domain corpus where we apply such methods is not big enough. With a distant supervised approach we train our classifiers in a domain-independent manner. The training corpus is created automatically by creating a domain-independent text corpus and leveraging the information contained in available knowledge bases. Once these classifiers have been trained to recognise relations in generic text, they can be successfully used in the concrete domain corpus, being the system able to extract domain-specific relations.

The method followed by the Learner Module[7] for applying a distant supervision approach for hypernym relations extraction is composed by the phases described below.

### 3.2.1 Training Phase

Before the Learner can extract relations from a given domain, it must have been previously trained using a large domain-independent corpus of text and one or more knowledge bases used as oracles. Using these elements, we perform the following steps:

- **Relational sentences corpus creation**. The relational sentences corpus is composed by a large set of sentences where two entities are related. In order to build such corpus in an automatic way:
  - We use the periodically available English Wikipedia dumps[8] as our initial English domain-independent textual corpus.

[7] https://github.com/epnoi/epnoi
[8] https://dumps.wikimedia.org/enwiki/

- For each sentence in each of the sections of each wikipedia article, the Learner inspects the entities that appear as term candidates (using the annotations provided by the NLP service) and creates all the possible pairs.
- For each of these pairs, the Learner checks whether there is a hypernymy relationship stored in the knowledge base between these entities making a query to a knowledge base containing all the previously generated pairs. If so, the sentence, along with the source and targets of the relation, is added to the relational sentences corpus.

In order to perform such data intensive task, the relational sentences corpus creator system has been implemented on top of the Spark[9] engine for large-scale data processing. A cluster of heterogeneous machines can be easily used to speed up such process.

– **Relational-patterns corpus creation**. For each sentence in the training corpus relation patterns can be automatically derived (note that one annotated sentence may generate several relation-patterns). Sentences are translated into relation-patterns because on the one hand it is helpful to abstract away from concrete subtleties that do not add meaningful information and burden the possibility of its generalisation to similar sentences; and on the other hand, to enhance the definitional sentence represented by the relation-pattern with additional linguistic information that provides much more information that their simple surface form. For each sentence the Learner generates one or several:

  - **Lexical relational pattern**. The set of lexical hypernym relation patterns are extracted creating a lexical hypernym relation pattern for each sentence in the relational sentences corpus, which implies performing the following actions:
    * We only consider a window of words composed by the words between the source and target of the definition, and with a size W to the left of the source term and to the right of the destination term. The rest of the words in the sentence are removed.
    * The beginning or the end of the sentences are marked with a special symbol in case that they belong to the considered window.
    * A linguistic annotation process is performed for each of the considered words in the sentence, and we annotate words with their part-of-speech grammatical category; and the source and destination (hypernyms candidates) of the definition are associated with their 7-class Stanford NER (i.e. Date, Location, Money, Organization, Percent, Person, Time) when possible.
    * All the words that are not verbs are substituted by their part-of-speech annotation, symbols are left unaltered, and the source/target terms in the definition are

[9] http://spark.apache.org/

replaced by the “<source>”/“<target>” token plus its NER category annotation. For example the sentence:

*The buildings are rendered by the 3D engine*

↓

ART **<source>Location** are rendered PREP ART ADJ **<target>Person**

- **Syntactic relational pattern**. The set of syntactic hypernym relation patterns are extracted creating a syntactic hypernym relation pattern for each sentence in the relational sentences corpus, which implies performing the following actions:
  - The same linguistic annotation process described above is performed for each of the considered words in the sentence, and we annotate words with their part-of-speech grammatical category; and the source and destination of the definition are associated with their 7-class Stanford NER (i.e. Date, Location, Money, Organisation, Percent, Person, Time)
  - A grammatical dependency annotation is performed in the sentence. We remove the directionality of its edges, transforming the sentence into an undirected graph; its nodes are the words plus their part/of-speech annotations, and its arcs binary grammatical relations.
  - Following the Shortest Path Hypothesis [7] all the words that do not belong to the shortest path between the source and target entities are discarded. In case that there is more than one shortest path, a pattern for each path is created.
  - All the words that are not verbs are substituted by their part-of-speech annotation, the source/target terms in the definition are replaced by the “<source>”/“<target>” token plus its NER category annotation.
  - Lastly, as [7] proposes, the negative polarity of verbs must be annotated (i.e. if we have a negation modifier, the prefix ’F’ is concatenated to the word surface form).

– **Classifier training**. Once the training corpus of definitional sentences has been transformed into a potentially bigger corpus of hypernym relation patterns, the latter is used to train two classifiers. These classifiers, once trained, will be able to recognise sentences that express hypernym relation between two entities of the sentence. More precisely we perform:

- **Generative lexical classifier training**. For recognising the lexical features along with the structure and ordering of definition sentences, we propose a generative approach. We create a probabilistic language model by analysing the set of lexical hypernym relation patterns that abstracts the training corpus. Once this probabilistic model is built, the generative probability of a given sentence, conditioned to such

probabilistic language model becomes the probability of the sentence of being definitional. In order to formalise the language model we use bigram model. Once we have trained the probabilistic language model using the set of lexical hypernym relation patterns, we can use the likelihood of new hypernym relation pattern to measure how likely it is to be a definitional sentence that signifies a hypernym relation. In other words, given a sentence form the domain corpus, and two of its terms, the probability of being a definitional sentence is given by the following expression:

$$P_h^l(\text{term}_i, \text{term}_j, s) = P(\text{hrp}^l(\text{term}_i, \text{term}_j, s)/\lambda) \quad (5)$$

  - $\text{hrp}^l(\text{term}_i, \text{term}_j, s)$ is a function that given a sentence $s$ and two terms ($\text{term}_i$, $\text{term}_j$), generates the corresponding lexical hypernym relation pattern.
  - $P(O/\lambda)$ is the probability of an observation $O$ given the language model $\lambda$ that we learnt using the training corpus.

- **Discriminative syntactic classifier training**. This classifier is trained using syntactic hypernym patterns from the relational sentences corpus, and once we have removed possible repeated patterns, each of these patterns becomes a feature for a logistic regression classifier. If we extract N distinct patterns, the feature vector would be an N-dimension vector; the i-th position represents whether the i-th syntactic hypernym relation pattern is matched or not. Using the training corpus, the N-dimension feature vector, and a 10-fold cross validation process we train the logistic regression classifier. For the learning algorithm we use Stochastic Gradient Descent (SGD)[10] since it has proved to be an efficient and suitable technique in the context of large-scale learning. The probability of a sentence of being definitional for two terms, from a syntactic perspective, is determined by the following expression:

$$P_h^s(\text{term}_i, \text{term}_j, s) = \frac{1}{1+e^{-\beta f(s)}} \quad (6)$$

  - $\beta$ is the regularised weight $N+1$-dimension vector trained using the SDG aproach.
  - $f(s)$ is a function that given a sentence $s$ returns an $N+1$-dimension vector. The i-th element is 1 or 0 depending on whether it matches the i-th syntactic hypernym relation pattern of the training set.
  - $f(s)$ first element is always 1.

### 3.2.2 Extraction Phase

Once the Learner has been trained, it can perform what we refer to as the extraction phase. Given a domain, in the extraction phase the Learner extracts the hypernym relations that appear in textual items of the domain. It translates into checking each sentence of the domain

[10] https://en.wikipedia.org/wiki/Stochastic_gradient_descent

to determine whether it is a sentence that expresses a hypernym relation or not. If a relation is found a new relation event is generated, containing both the found relation, its estimated probability, and the sentence itself (becoming the hypernym relation provenance sentence). More specifically, for each sentence $s_k$ that belongs to the domain specific corpus the Learner carries out these actions:

- The set of terms candidates of the sentence are obtained using the very same transducer described for the terminology extraction.
- The probability of this sentence of being definitional for each pair of terms selected from the terms candidates is calculated as the linear combination of the probabilities obtained using the generative lexical classifier and the discriminative syntactic classifier.

  Using the same notation as in the training phase, this probability can be expressed as follows:

$$P_h = \alpha * P_h^S(\text{term}_i, \text{term}_j, s_k) + \beta * P_h^l(\text{term}_i, \text{term}_j, s_k) \quad (7)$$

In case that this probability is greater than a configurable threshold the hypernym relation is considered as found between these two terms in the given domain; and a created relation event is fired then as a consequence.

### 3.3 Implementation

The learner implemented in DRInventor platform is available on Github at the following Repository: `https://github.com/epnoi/epnoi`. The code is organised in different modules, such as the NLP toolkit[11], the data store[12], the knowledge base[13] for distant supervision, or the learning module[14].

The whole learning approach is implemented by leveraging on different open source frameworks for a better performance and reusability. Since most of the learning process is very demanding in term of computational resources (generation of a big number of patterns from the sentences in the textual collection, search and manipulation of such patterns and annotations in order to infer new ones) we have relied on Spark[15] for parallelising the different annotation phases along the different documents. Storing such a big number of sentence patterns, together with the terms and relations, requires specific databases able to deal with the cardinality of such intermediate knowledge bases. In our case we have relied on Casandra[16]. For

[11] `https://github.com/epnoi/epnoi/tree/develop/nlp`
[12] `https://github.com/epnoi/epnoi/tree/develop/store`
[13] `https://github.com/epnoi/epnoi/tree/develop/knowledgebase`
[14] `https://github.com/epnoi/epnoi/tree/develop/learner`
[15] `http://spark.apache.org/`
[16] `http://cassandra.apache.org/`

easing the storage of in-memory Java structures, we have also used MapDB[17]. Finally in order to expose the information in the form of triples and be able to apply SPARQL queries over the graph of terms and relations, this information is also available in a Virtuoso[18] triple-store.

### 3.3.1 Accessing the Ontology Learning Results via DRInventor API

The DR Inventor REST API[19], offers different methods for retrieve both the *Terms* and the *Relations* that have been found over the corpora of documents indexed in the platform by applying the approach described in section 3.2 and 3.1.

Retrieve the complete list of terms generated and stored in the system is straightforward. We only need to invoke the following GET method:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/terms/
Parameters:
Method: GET
Response:
[
  "http://drinventor.eu/terms/eefec303079ad17405c889e092e105b0",
  "http://drinventor.eu/terms/67e92c8765a9bc7fb2d335c459de9eb5",
  "http://drinventor.eu/terms/1f09edca718cff07e4fc0d8ffa8f3303",
  "http://drinventor.eu/terms/36f0ff15dcdecfecdd8cf092457be7d",
  "http://drinventor.eu/terms/ef72c37be9d1b9e6e5bbd6ef09448abe",
  ...
]
```

We can get further information about a particular term by performing another GET request specifying the UUID of the particular term we want to check out, in particular the content of term (the surface form) and the normalised score provided by the ontology learning approach, indicating its importance.

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/terms/9bc81c3aa886b690f84c5aba4109e20
Parameters:
Method: GET
Response:
{
  "uri": "http://drinventor.dia.fi.upm.es/terms/9bc81c3aa886b690f84c5aba4109e20",
  "creationTime": "2016-08-29T15:17+0000",
  "content": "Pixel",
  "score": 0.872
}
```

We can retrieve the list of relations in a similar manner. In order to get the full list of hypernyms we can rely on the following method:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/relations/
Parameters:
Method: GET
```

[17] http://www.mapdb.org/
[18] http://virtuoso.openlinksw.com/
[19] http://drinventor.dia.fi.upm.es/api/

```
Response:
[
  "http://drinventor.eu/relations/63bcabf86a9a991864777c631c5b7617",
  "http://drinventor.eu/relations/f7b44cfafd5c52223d5498196c8a2e7b",
  "http://drinventor.eu/relations/574ff4699083ce51de0dabcfad5edc4c",
  "http://drinventor.eu/relations/8ac20bf5803e6067a65165d9df51a8e7",
  "http://drinventor.eu/relations/eb399bcaca686f8609137153307eecf1",
  "http://drinventor.eu/relations/9a8c2b9d518bc163e99611fbacea63b2",
  "http://drinventor.eu/relations/f8b0b924ebd7046dbfa85a856e4682c8",
  "http://drinventor.eu/relations/8d5f9e9048e2000531c3170f4b833b1",
  "http://drinventor.eu/relations/8512ae7d57b1396273f76fe6ed341a23",
  "http://drinventor.eu/relations/441f9e2d94c39a70e21b83829259aa4",
  "http://drinventor.eu/relations/1e4483e833025ac10e6184e75cb2d19d",
  "http://drinventor.eu/relations/d7c95dd61cc3588432f3b3eef94101e9",
  ]
```

In order to access the details about a particular relation, we can perform a GET request by specifying the UUID of the relation as a parameter in the URL. The results will show the source and target terms that hold the hypernym relation, and a confidence score that indicates the system's level of certainty when spotting such relation.

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/relations/8d5f9e9048e2000531c3170f4b833b1
Parameters:
Method: GET
Response:
{
  "uri": "http://drinventor.dia.fi.upm.es/relations/8d5f9e9048e2000531c3170f4b833b1",
  "creationTime": "2016-08-29T15:17+0000",
  "source": "http://drinventor.eu/terms/4efa264f5ef3e1a5c95736e07544ebf0",
  "target": "http://drinventor.eu/terms/d6fe1d0be6347b8ef2427fa629c04485",
  "score": 0.215
}
```

In addition we can obtain all the terms that are paired via hypernym relations to a specified term $T$ with UUID $id$ (for example, b891b62ab9be7813b9c97aec94a62fff), by invoking the method below:

```
Url: http://drinventor.dia.fi.upm.es:80/api/0.2/words/b891b62ab9be7813b9c97aec94a62fff/words

Parameters:
Method: GET
Response:
{
  "uri": "http://drinventor.dia.fi.upm.es/relations/8d5f9e9048e2000531c3170f4b833b1",
  "creationTime": "2016-08-29T15:17+0000",
  "source": "http://drinventor.eu/terms/4efa264f5ef3e1a5c95736e07544ebf0",
  "target": "http://drinventor.eu/terms/d6fe1d0be6347b8ef2427fa629c04485",
  "score": 0.215
}
```

### 3.3.2 Exposing Ontology Learning Results via Semantic Web Technologies

Ontologies constitute a very powerful way of formalising the knowledge about the Web. Even they can be materialised in very different ways, we advocate the use of open Web standards that can help us to provide the generated model according to the Semantic Web principles. The results of the DRInventor learning algorithm are available following some of those standards in two different ways:

1. Via a SPARQL endpoint. SPARQL[20] is a graph-based query language for RDF. The information inferred by the learner is stored in a dedicated triplestore (a Virtuoso instance, as mentioned before) that can be accesible in order to launch queries that retrieve both terms and relations. In the case of the SIGGRAPH corpus, you can access an instance of the graph by pointing to the URL `http://wiener.dia.fi.upm.es:5090/sparql`

2. Via an OWL/RDF serialisation. OWL[21] is an ontology language for the Semantic Web with formally defined meaning, providing a way of specify classes, properties, individuals and data values. Therefore the same terms and relations that are stored in the Virtuoso triple-store can be exposed in this format for other Web agents to consume them. Below we show an excerpt of some terms and relations extracted from the SIGGRAPH corpus and serialised as OWL classes in Turtle[22] syntax.

```
@prefix rdf:     <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:    <http://www.w3.org/2000/01/rdf-schema#> .
@prefix oa:      <http://www.w3.org/ns/oa#> .
@prefix owl:     <http://www.w3.org/2002/07/owl#> .
@prefix epnoi:     <http://www.epnoi.org/ontology#> .
@prefix xsd:     <http://www.w3.org/2001/XMLSchema#> .

Papers:
<http://drinventor.eu/documents/11b345589fbfde1785585d7c342d6cd0> a epnoi:Paper .
<http://drinventor.eu/documents/6dfa0fa495d717d421932692b10f753d> a epnoi:Paper .
<http://drinventor.eu/documents/e0cc14da6195656edc0ce84cc146189e> a epnoi:Paper .

Terms:
<http://drinventor.eu/terms/2e496adbe3e5b8080f2dc95548254608/touch_sensor>
  a epnoi:Term , epnoi:Annotation .
<http://drinventor.eu/terms/70eca1b14e33a948487a81d049a5b30b/standard_painting>
  a epnoi:Term, epnoi:Annotation .
<http://drinventor.eu/terms/db59fd94942cf8d6b2e08560446162c4/face>
  a  epnoi:Term , epnoi:Annotation ;
  rdf:label "face" ;
  oa:annotatesDocument <http://drinventor.eu/documents/e0cc14da6195656edc0ce84cc146189e> ;
  epnoi:relevance "0.854"^^xsd:float .

Relations:
```

[20] https://www.w3.org/TR/rdf-sparql-query/
[21] https://www.w3.org/TR/owl2-primer/
[22] https://www.w3.org/TR/turtle/

```
http://drinventor.eu/relations/34a844ae0bdc268415a4c37198aefda4
  a epnoi:Relation , epnoi:Annotation .
http://drinventor.eu/relations/130aed1c8ebf5cbd14db363e70b44bc5
  a epnoi:Relation , epnoi:Annotation .
http://drinventor.eu/relations/7bca6e0a8803f23f5fe01e250aab0356
  a epnoi:Relation , epnoi:Annotation ;
  oa:annotatesDocument <http://drinventor.eu/documents/e0cc14da6195656edc0ce84cc146189e> ;
  oa:source <http://drinventor.eu/terms/db59fd94942cf8d6b2e08560446162c4/face> ;
  oa:target <http://drinventor.eu/terms/70eca1b14e33a948487a81d049a5b30b/standard_painting> ;
  epnoi:relevance "0.487"^^xsd:float .
```

Finally we include some figures about the SIGGRAPH corpus in order to better understand the the magnitude of the indexing and annotation processes. The total number of papers indexed in the platform and therefore considered in the knowledge graph is **1514**. The total number of terms found by the DRInventor learning approach is **75874**. The number of hypernym relations between pairs of concepts is **78902**. This huge number of instances is not manageable for final agents consuming the model, so the API and any agent querying the annotations can filter them by relying on the relevance score that they have attached. In section 4 on evaluating the terms and relations, we will show how we only keep the top 100 terms for the recommendation and discovery operations.

# 4 Evaluation of Ontology Learning Methods

As part of the efforts made in DRInventor for improving our ontology learning approach, we have also focused on the different strategies that may be used for evaluating their results. Establishing a function able to determine the success of an ontology learning approach is a very complex task even for domain experts, since it can involve many considerations and requieres a strong background knowledge in order to do fair judgements over candidate terms and relations. However, evaluation is a crucial step in any learning process, because of the two reasons already stated in the executive summary: 1) it allows comparing different approaches in a systematic way, and 2) it gives support to learning approaches relying on iterative searches in the space of solutions, which need functions to determine how close they are to the optimal result.

## 4.1 The Ontology Learning Evaluation Task

We first popose a set of definitions that will be useful to describe the objectives and design decisions in our ontology evaluation methodology.

### 4.1.1 A Definition of Ontology for Automatic Learning Tasks

Below we formalise the concept of an ontology that aims to better match the specific requirements of the ontology learning domain. In contrast with much more complex formalisations of ontologies [13], this research topic relies on less constrained models that are automatically built following principles such as incremental generation (the schema evolves as more items in the corpora are processed) or flexibility (knowledge can change when more items are added into the corpora). According to this, we introduce the definition of a flatten ontology as a triple composed by the sets, $W$, $R$, $P$:

$$\mathcal{O} = \{W, [R], [P]\} \tag{8}$$

**Definition 1.** *Flatten Ontology Simplified representation of an ontology considering three different sets: the terms $W$, the set of relationships between terms $R$, and the global metadata properties $P$. It focuses in the functional dimension of the ontology, leaving aside some structural details that are not relevant in recent automatic learning tasks.*

The first set of terms in $W$ is defined as $\forall w \in W, w \in \mathbb{S}$, being $\mathbb{S}$ the set of all strings generated as a combination of letters in our alphabet. $R$ is the set of all the relations established between pairs of terms $w_a$ and $w_b$, formalised as a triple $\forall r \in R, r = \{w_a, w_b, c\}$, where $c \in \mathbb{S}$ specifies the kind of connection established between $w_a$ and $w_b$. The last set $P$ defines the different metadata properties that sometimes are further characterising certain ontologies: general description, list of keywords, etc. Each property $p \in P$ is a pair of name $n$ and a string value $v$. It is important to note that, as the name suggests, this definition of Flatten Ontology leaves out

certain information that can be explicitly present in more exhaustive content representation structures considering for example restrictions bounding property domains and ranges. They have been left aside given they are not relevant in the current horizon of ontology learning techniques. The set of relations $R$ and metadata properties $P$ may be empty in order to keep the formalisation flexible enough for different learning tasks.

### 4.1.2 Ontology Learning Evaluation Objectives

Trying to capture the knowledge about a particular domain or subdomain can result in heavy-weight formalisations with many axioms or restrictions, which may be expressed in formal languages like OWL[23]. However not every task leveraging on ontologies requires this degree of specificity. On the one hand, real world applications rarely need very complex representation models, because they are too complicated for expert users and developers. On the other hand, ontology learning techniques employ many state-of-the-art approaches that are still far from being able to deal with details such as cardinalities or universal quantifications.

In order to better address this complexity, in this research work we introduce the notion of *Evaluation Objectives* $\omega$:

**Definition 2.** *Ontology Learning Evaluation Objective. Given an ontology* $\mathscr{O}$*, an evaluation objective* $\omega$ *is a particular subset of* $\mathscr{O}$ *that we aim at evaluating. Given the formulation of a Flatten Ontology previously presented in Equation 8, those evaluation objectives can therefore be the lexical* $W$*, taxonomical* $R$*, or general metadata* $P$ *layers, or all their possible subsets and combinations.*

Depending on the objectives established on each ontology learning initiative, those evaluation objectives can vary from the more basic, lexical-oriented goals, to others putting emphasis on the relations between the identified concepts.

### 4.1.3 Ontology Learning Evaluation Methods

Having considered a set of evaluation objectives $\omega_1, \omega_2, ..., \omega_n$ we need different methods to determine how good those features were learnt. Hence we introduce the definition of *Evaluation Method* $\mathscr{F}$:

**Definition 3.** *Ontology Learning Evaluation Method. Given an automatically generated ontology* $\mathscr{O}$*, and an evaluation objective* $\omega$*, an Evaluation Method is a function* $\mathscr{F} : \omega \rightarrow \mathbb{R}$ *that expresses into a unified score the degree of success of the learning algorithm for automatically generating* $\omega$*.*

We consider as the ideal output of the learning approach, what an unbiased set of expert human annotators with infinite amount of time would have chosen as candidates after analysing the entire corpus.

[23] https://www.w3.org/TR/2012/REC-owl2-syntax-20121211/

**Quality Criteria for Evaluation Methods**

Besides the good principles already reported in the literature for measuring different information tasks and systems [26], below we identify a set of desirable characteristics that ontology learning methods should follow in order to better serve their purposes. They are grounded on the very particular needs of the ontology learning task, which demands very changeable and easily formalisable knowledge models, commonly generated by iterative and train-based algorithms that may require the method $\mathscr{F}$ to be executed multiple times.

- Cost-effective. Algorithms learning patterns from data are often based on supervised approaches [34] that require the evaluation to be performed multiple times over a particular subset of the data, hence the method has to be affordable to execute in terms of time and resources.
- Reproducible. The method $\mathscr{F}$ has to be easy replicable not only by a specific learning approach but also by other systems being developed a posteriori.
- Extensible. The function $\mathscr{F}$ should be relatively easy to update in case that new considerations or observed facts are later considered, in order to offer the maximum level of trustiness.

We decide to systematically discard qualitative-oriented methods, which are much more subjective, difficult to define, and complicated to interpret.

**Types of Evaluation Methods** Evaluation methods $\mathscr{F}$ can be implemented following very different philosophies and techniques. In the list below we introduce some of the most relevant ones, inspired on a similar classification in [12].

- Task-based ($\mathscr{F}_{Task}$). This kind of evaluation methods try to measure how much a system improves in performing a certain task when an ontology is integrated into its workflow [27]. One of the most representative examples of such evaluation method is available at [27]. The problem with this perspective is that methods are so specific that it is complicated to find well-suited measures to be applied. Also, it is influenced by implicit factors that make it harder to solely attribute the improvements to the use of a certain ontology.
- Criteria-based approach ($\mathscr{F}_{Criteria}$). In this case certain expected patterns, properties and rules are set beforehand and checked over the results of the algorithm being tested [20]. An example of this kind of evaluation method can be found in [20]. This kind of techniques are very appropriate for programatic evaluations, but they are sometimes difficult to interpret and justify so they have to be further supported by other evaluation methods.
- Corpus-Based ($\mathscr{F}_{Corpus}$). They are good in measuring functional aspects, easy to automate and reproducible by third parties. However they assume that the corpus used as ground truth is representative enough of the domain, so the process of generating such

dataset can end up being significantly tedious and exhaustive. They normally involve the use of information extraction and knowledge representation techniques already existing in the literature, which need to be well-matured and stabilised in the domain while independent of the learning algorithm that are being the subject of the evaluation. Some examples of this methods are described in [11].

– Assessment ($\mathscr{F}_{Assess}$). Experts in the domain or potential consumers of the results go through the output of an algorithm to judge on their validity. This is the most intuitive way of implementing an evaluation method, at the risk of not being able to define a clear set of guidelines that align all annotators into a well defined task and the difficulty to recreate certain conditions to fairly compare the current approach with other systems. This normally requires the humans annotators to completely re-do the annotation over the whole corpora every time a change is performed. In addition, certain conditions are impossible to reproduce so even with a very committed set of workers the results between different executions can not be compared.

**The Cost of an Evaluation Method** Going deeper into the criteria introduced above, one of the most important aspects to be taken into account when selecting one particular evaluation method against others is the cost $\mathscr{C}(\mathscr{F}) : \mathscr{F} \rightarrow \mathbb{R}$ of executing it. It may happen that a method is highly reliable in determining the quality of an ontology learning algorithm, but the cost of execution is so high that researchers will be discouraged to use it, moving to less desirable practices instead. Without aiming to be exhaustive and just to emphasise the existence of different temporal and resource-based costs associated to each evaluation method $\mathscr{F}$, we identify two different kinds of costs: the one associated to techniques involving humans in the evaluation, $\mathscr{C}(\mathscr{F}_{Human})$, and the one derived from the automatic execution of an algorithm $\mathscr{C}(\mathscr{F}_{Automatic})$. For the sake of simplicity, we consider that the human costs associated to crowdsourcing campaigns (higher number of annotators, average/low knowledge about the domain) are pretty much the same than the ones relying on experts (lower number of annotators, deep knowledge about the domain). We also differentiate between a creational process where annotations are being generated from scratch by humans, and a validation process where some already existing learning results are just being judged: costs, $\mathscr{C}(\mathscr{F}_{Create})$ and $\mathscr{C}(\mathscr{F}_{Assess})$. We establish two main premises that will influence our later decisions in the ontology learning evaluation methodology described in section 4.2:

$$\forall \mathscr{F}_i, \mathscr{C}(\mathscr{F}_{Automatic}) \ll \mathscr{C}(\mathscr{F}_{Human}) \tag{9a}$$

$$\forall \mathscr{F}_i, \mathscr{C}(\mathscr{F}_{Assess}) \leq \mathscr{C}(\mathscr{F}_{Create}) \tag{9b}$$

### 4.1.4 Evaluation Methodology

Having described the concepts of ontology learning evaluation objective $\omega$ and ontology learning evaluation method $\mathscr{F}$, we finalise by formalising the notion of an evaluation methodology

$\mathscr{M}$. Given that for each $\omega_i$ we need a evaluation method operating over it, we also introduce the concept of dimension $d = (\omega_i, \mathscr{F})$ as a way to make explicit this pair.

**Definition 4.** *Evaluation Methodology. An evaluation methodology is a list of evaluation dimensions $d$, formalised as a pair $(\omega_i, \mathscr{F})$ where for each evaluation objetive $\omega_i$ identified there is a corresponding $\mathscr{F}_i$ that allows evaluating it.*

$$\mathscr{M} = \{d_1, d_2, d_3, ....d_n\} \tag{10}$$

## 4.2 An Integrated Ontology Evaluation Approach

In this section we present our unified vision on the evaluation of ontology learning algorithms. We intend to cover the most relevant use cases being considered in the field, from the more relaxed thesaurus-oriented approaches that are only concerned about finding sets of terms, to the more tightly constrained efforts that also consider relations between those terms, or even global indicators describing the ontology as a whole. In addition and as stated in the introduction, we start from an ontology learning scenario where the diversity and changeability of the datasets make it difficult to find a good ontology that maches the underlying data.

In this deliverable we introduce a new ontology learning evaluation methodology $M_{OntoLearn}$ that aims to normalise the evaluation procedure along the different research efforts in the field.

Our contribution in this research problem is twofold:

1. We propose three evaluation dimensions $d_W$, $d_R$ and $d_P$ targeting different aspects represented in our definition of *flatten ontology* introduced in section 4.1. This way we integrate into a single methodology different needs from the ontology learning community.

2. We propose an evaluation method called Hybrid-GS (denoted as $F_{Hybrid}$) to be applied on at least the two first dimensions $d_W$ and $d_R$. This methodology extends the corpus-based methods already applied in the literature [11] [38], through the application of principles to ensure good coverage and similar quality to pure human-driven approaches.

Therefore, and taking advantage of the definitions in section 4.1 and the notion of flatten ontology $\mathscr{O}$ we define our methodology as follows:

$$\mathscr{M}_{OntoLearn} = \{(\omega_W, F_{Hybrid}), (\omega_R, F_{Hybrid}), (\omega_M, F)\} \tag{11}$$

Fig 2 depicts the whole integrated evaluation process, presenting the 3 different dimensions proposed $M_{OntoLearn}$ and how the $F_{Hybrid}$ method can be used to evaluate at least dimensions 1 and 2.

We observe how the columns in the figure correspond to the three research objectives on which our methodology is based: terms, relations, and general properties. Most of the approaches in the literature target at least the first dimension since they focus on identifying a

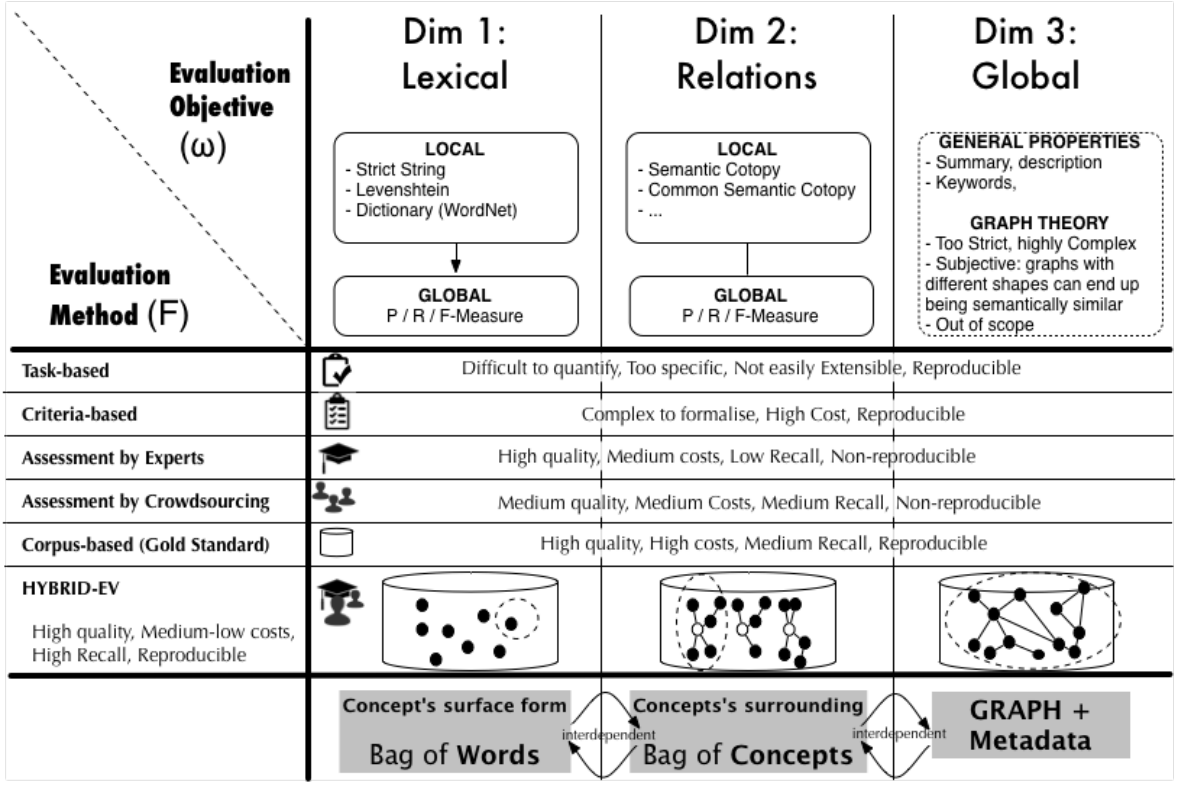


Figure 2: General view of the Evaluation Methodology

set of terms, while remaining agnostic to connections between them. The second dimension tries to determine the adequacy of the taxonomic and non-taxonomic relations established between terms. The third and last dimension considers the ontology as a whole, sometimes using graph-based measures, so that it can be difficult to evaluate and interpret. It can also involve different global properties that are easier to compare, but hardly reused by ontology learning tasks. The rows in figure 2 represent the different evaluation methods $\mathscr{F}$ that may be used for evaluating the aforementioned research objectives. The reasons that led us to propose a new method $F_{Hybrid}$ were mainly: 1) task-based approaches are difficult to quantify and specific, and the improvements in the scores cannot be straightforwardly linked to an increase in quality of the learned ontology, 2) criteria-based methods are complex to formalise, justify and maintain, 3) assessment from experts and crowdsourcing campaigns provide good quality but normally low recall and reproducibility, and 4) traditional gold-standard approaches offer good quality annotations but incur in significant costs and do not necessarily ensure high coverage.

### 4.2.1 The Method Hybrid-GS

We introduce a novel evaluation method that is inspired by the corpus-based evaluation methods [11], but introducing various advantages with the objective of reducing the cost of generating a gold standard while maintaining an adequate quality in the generated annotations. The so called $F_{Hybrid}$ method aims to leverage on automatic annotation techniques as much as possible, while still performing a human assessment on top that is ideally more lightweight than creational processes (see equation 9b) and performed only once. This way, we get the best of the three worlds: corpus-based methods that are highly reproducible, automatic approaches that can be executed at a lower cost (see equation 9a) and bring higher coverage by quickly processing large sets of documents in short time, and the precision of human assessments, especially in user oriented tasks.

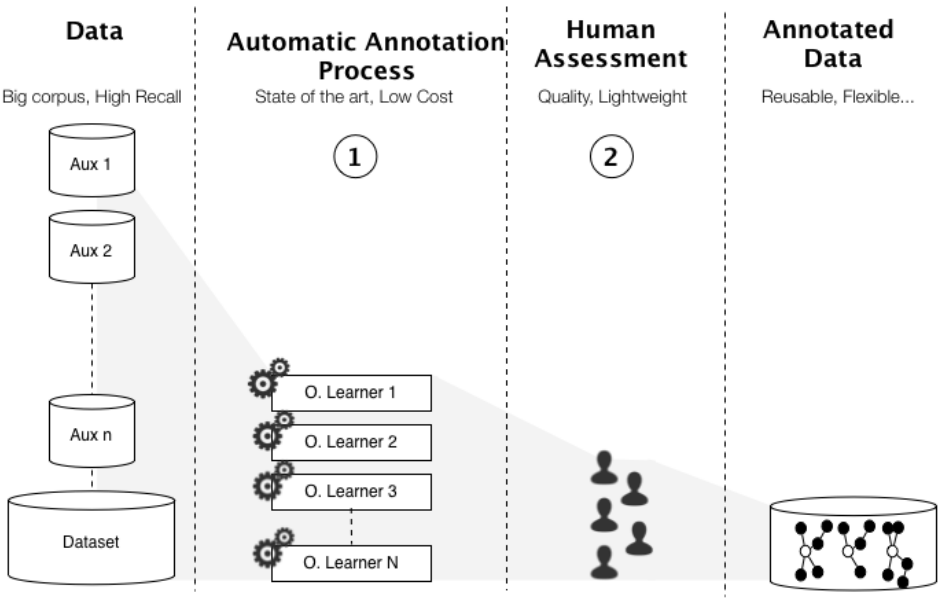


Figure 3: Data selection flow in $F_{Hybrid}$ evaluation method

The method $F_{Hybrid}$ is composed by two selection phases, labeled in Figure 3 as *Automatic Process* and *Human Assessment*. In the first one, different state-of-the-art automatic learning techniques (ideally more than one and never the one being benchmarked) are executed in parallel to produce a first set of annotations. Since this first set of annotations is automatically generated, we can leverage on more than one dataset in the domain $Aux_1$, $Aux_2$, ... $Aux_n$ in Figure 3. The second phase is performed by humans and consists in validating the integrated results from the ontology learners. Thanks to the use of automatic techniques in the first step, the original corpora being annotated can be accompanied with other relevant datasets, significantly increasing the coverage of the results. To better summarise those particularities and complement the hypothesis in equation 9, and being $\mathscr{C}$ the cost of executing a particular evaluation function $\mathscr{F}$ we formulate the following inequalities:

$$\mathscr{C}(\mathscr{F}_{Human}) \ll \mathscr{C}(\mathscr{F}_{Hybrid}) \leq \mathscr{C}(\mathscr{F}_{Auto}) \tag{12a}$$

$$Prec(\mathscr{F}_{Human}) \approx Prec(\mathscr{F}_{Hybrid}), Rec(\mathscr{F}_{Auto}) \approx Rec(\mathscr{F}_{Hybrid}) \tag{12b}$$

#### 4.2.2 Dimension 1: Lexical

This dimension $(\omega_W, F_{Hybrid})$ has as ultimate objetive $\omega_W$ to evaluate the lexical aspect of the ontology being built. We focus on verifying whether the set of terms $t$ automatically generated correspond to what an unbiased set of expert annotators with infinite time would have chosen after analysing the entire corpus.

Therefore the evaluation objetives will come under the form of a bag of terms $\{t_1, t_2, ...t_n\}$ if the order does not matter, a list $(t_1, t_2, ...t_n)$ if they are ranked in importance, or similar aggregations. For evaluating the quality of those results one solution is to apply traditional Precision and Recall methods over the whole set of terms automatically extracted (labeled as "Global" measures in Figure 2), by comparing them with the Gold Standard ideally created by methods $F_{Hybrid}$, which contains the set of $W$ as specified in the definition of flatten ontology. In cases where order matters, other measures from information retrieval can be used, such as *Mean Average Precision* (MAP) or *Normalised Discounted Cumulative Gain* (NDCG) [10]. We would like to emphasise that some of those measures are too focused on the performance at the top positions of the automatically retrieved list, therefore neglecting the big picture of the domain vocabulary that we are trying to generate programatically. Other measures more oriented to coverage are normally preferred since today's data exploitation tasks tend to prioritise representativeness of the learned information unit against very high precision (see special measures such as *Compactness* reported in [18])

To obtain this global score we need a local similarity measure to operate between pairs of items from both the ontology learning results and the Gold Standard. We consider two kinds of distances, 1) the ones relying solely on the terms' surface form (Strict Distance and Relative String Distances[24] such as the Jaro-Winkler distance), and distances leveraging also on external knowledge sources for further improving the comparison, such as WordNet [1] or DBpedia [23].

#### 4.2.3 Dimension 2: Relations

This dimension $(\omega_R, F_{Hybrid})$ has as ultimate objetive to check the potential relations established between the previously identified terms $(\omega_R)$, whether they are taxonomic (hyponymy and hypernymy) or not (thematic). Unlike other approaches that try to decouple the dependency between the lexical and the relational levels [12], we acknowledge this singularity and assume it as an inherent characteristic of the ontology constituents. To capture this notion of relations,

[24] https://en.wikipedia.org/wiki/String_metric#List_of_string_metrics

we focus on verifying what we call concept $c$, defined as a subset of relations $r_c \in R_c$ and terms $w_c \in W_C$ such that there exists a set of $n$ relations in and $n-1$ terms that connect $w_c$ with the seed term $t_c$, being $n$ the maximum depth considered. Similar approaches leveraging on local neighbourhoods have been introduced in other research work like [37].

The evaluation objetive will come under the form of a bag of concepts $\{c_1, c_2, ...c_n\}$, or similar aggregations. As in the previous dimension, we will be mainly interested in applying traditional Precision and Recall methods over the whole set of concepts by comparing them with a Gold Standard, ideally created by following the method $F_{Hybrid}$ but now adding also relations $R$ to the list of terms $W$. For implementing the local measure, there are different distances $D(c_a, c_b) \rightarrow \mathbb{R}$ in the literature that rely on the surrounding terms and relations [37]. Here we highlight two different measures called *semantic cotopy* and *common semantic cotopy*, as described in [11]. It is worth noting that all the terms and relations between concepts belong to the ontology being learnt and not to external sources like WordNet or DBpedia, which fall into the lexical dimension.

#### 4.2.4 Dimension 3: Global

The last considered dimension looks at the big picture of the ontology by analysing it as a whole. The objective $\omega_P$ takes into account global properties of the ontology such as topics, keywords, or summaries that can be generated via some corpus summarisation approaches, but it can still highly leverage on the set of terms $W$ and properties $P$, revealing once again how dimensions are incremental in complexity but interconnected with previous levels. These kinds of methods are far more complex to apply, and this is why we haven't specified a particular kind of function in the definition of our methodology $\mathscr{M}_{OntoLearn}$. However $F_{Hybrid}$ methods can be equally applied, finally getting to construct an entire *flatten ontology* that can be used as a Gold Standard. We have identified three kinds of global-oriented evaluation methods:

1. As the set of terms and relations are shaping up a graph, we can check on its isomorphism against the reference ontology. In this particular case, we may re-use the same Gold Standard generated for evaluations considering dimension 2 that already includes nodes and connections. Examples of such similarity functions can be found at [17] where the authors rely on a graph edit distance. The problem of this kind of evaluations is that the morphological differences between the two graphs do not necessarily correlate with the cognitive gap between them.

2. *keywords and topics* can be part of the set $P$ included in the definition of flatten ontology since they have a "global" scope. We can create them by applying a $F_{Hybrid}$ method, as we did with terms. However they are discouraged to be used in isolation, since they focus too much on representativeness and leave aside other details that are essential in the learned ontology.

3. The last alternative leverages on *summaries*, which equally have a global nature. The main drawback of such summaries is that they normally focus on describing involved agents (instances, entities), but ignore more fine-grained ontology details shaping the kind of knowledge available in the corpus. They can be compared against ground-truth summaries, but more affordable $F_{Hybrid}$ evaluation methods are not applicable since in most cases, summaries have to be created from scratch.

As we will see in Section 4.3, most of the ontology learning approaches only consider the first and second dimensions, due to the already-discussed complexity for the third dimension. In some particular evaluation tasks we can consider to combine the 3 different dimensions into a single score that gives a general idea of the ontology learning technique, bearing in mind that some of the phases are optional.

### 4.3 $M_{OntoLearn}$'s Use Cases in the Literature

Having described our methodology $M_{OntoLearn}$, in this last section we present an analysis of some ontology learning approaches in the literature and the way they have been evaluated, in order to study 1) the way they overlap with our integrated methodology 2) how they can benefit from a better formalised evaluation strategy and the less human-dependent, quality-focused evaluation methods introduced in this paper. The three selected systems are sorted in increasing order of complexity, according to the dimensions in $M_{OntoLearn}$ they target.

**Case 1: Learning Domain Ontologies for Web Service Descriptions**. In this research work [30] the authors apply their approach over an experimental corpus consisting of 158 EMBOSS bioinformatics service descriptions. Their evaluation looks into the *first dimension* $d_W$ identified in our approach, to measure the lexical precision of the generated ontology by comparing the results against a set of ground truth annotations manually identified from the corpus. This evaluation would have benefited from the application of a $\mathcal{F}_{Hybrid}$ method leveraging on already existing term-spotting techniques, in order to reduce the human intervention to a less demanding assessment phase.

They tried to tackle also properties (the *relations dimension* $d_R$), but they found out that de-facto ontologies were too much complex and very different from the extracted ontology. This backs our claim on needing to move from very formal high-level ontologies to more lightweight models. Hence, they put domain experts to rate concepts according to their usefulness in the current task. However, this is not reproducible so other approach working on similar data sets would not be able to reuse such efforts.

**Case 2: OntoLearn**. OntoLearn [35] evaluation strategy is twofold: first, they provide a detailed quantitative analysis of the ontology learning algorithms, mainly focused on the lexical aspect ($d_W$). Secondly, they automatically generate natural language descriptions of formal concept relations in order to facilitate qualitative analysis by domain specialists, therefore targeting $d_R$.

They claimed that a manual analysis of the extracted terminology is advisable before proceeding with the subsequent steps, arguing that this task lasts about 0.5 minutes per term so it can be easily accomplished in few hours by domain specialists. Those figures support our hypotheses in equation 9b. But unfortunately their assessment is directly made over the results of the algorithm being evaluated, so it cannot be straightforwardly used for developing $\mathscr{F}_{Hybrid}$-based methods that require to leverage on various state-of-the-art automatic algorithms to ensure unbiased results. Concerning the evaluation of relations, they developed a "gloss" generation algorithm in order to facilitate per-concept evaluation by domain specialists. The objetive is to reduce the cost of human assessments, but since they are directly performed over the results they again become non reproducible for future campaigns. Other examples of systems considering evaluations fitting into dimensions 1 and 2 are [11] and [24].

**Case 3: CRCTOL**. The approach presented in [22] has been already introduced in section 2, as one of the most complete evaluations available in the literature. Having now described our methodology $M_{OntoLearn}$, we can study how their evaluation objetives fall into our proposed dimensions.

They use the concept "component level" to refer to the lexical aspect expressed by dimension ($d_W$). They also consider some so-called "relations", including taxonomic and non-taxonomic ones, therefore covering the whole spectrum of connections targeted by dimension $d_R$. They evaluated these two first dimensions by relying on a manually annotated corpus coming form the US report "Patterns of Global Terrorism Documents", from 1991 to 1994. The problem lays again within the high temporal costs of performing such generation process by relying only on experts in contrast to $\mathscr{F}_{Hybrid}$-like methods. In addition, CRCTOL has been also evaluated in what they named the "Structural Property Based Method". Based on the "small world property" [28] that applies to knowledge networks such as WordNet, they assume that their automatically built domain ontology should also fit this principle. Hence, they gauge the quality of the built ontology by measuring whether its graph representation is consistent with that of a small world graph. This graph-based evaluation technique has a global scope that allows to classify it within the methods in the third dimension $d_P$ of the methodology.

The organisers of Semeval-2015 [6] also targeted dimension 3 via some structural indicators such as the size of the taxonomy in terms of nodes and edges, the degree of connectivity with the root, and the existence of cycles. Part of their evaluation is very much in line with our $\mathscr{F}_{Hybrid}$ method: taking as input the results submitted by the participants (with affordable $\mathscr{C}(\mathscr{F}_{Auto})$), they asked experts to identify relations which were initially missing in the gold standard (lightweight human intervention $\mathscr{C}(\mathscr{F}_{Assess})$) in order to increase coverage.

### 4.4 $M_{OntoLearn}$ for Evaluating DRInventor Learning Techniques

Having explained the methodology $M_{OntoLearn}$, which tackles the problem of evaluating ontology learning approaches, in this subsection we show how to apply it over the computer graphics

domain through the corpus and features available in DRInventor Platform. We will introduce some peculiarities of this domain, the experts that have been involved in the study, the different tools we have leveraged on for performing the evaluation, and the finals results focusing specially on the lexical layer of the ontology.

As introduced before, the direct use of domain experts to generate a *Gold Standard* incurs in significant implementation costs. Through the use of this methodology, we have been able to perform an initial evaluation that reduces the amount of human intervention needed. Concerning **Dimension 1**, we have implemented a lightweight evaluation based on experts' feedback: $\mathscr{F}_{Assess}$. However and given the complexity of the relation learning task, which has not been deeply tested yet, for **Dimension 2** we have only performed some manual assessments as a preliminary check before a more exhaustive application of the methodology and the consideration of measures concerning **Dimension 3**.

### 4.4.1 The Domain of "*Computer Graphics*"

The corpus of research objects indexed in DRInventor Platform[25] contains a set of scientific papers in the domain of computer graphics, which have been indexed (in the form of resources at different levels of granularity, such as Documents, Parts, Items, etc) and annotated following techniques such as Topic Modelling as described in deliverable 5.6 [29].

As already mentioned, the selection of the human experts that will take part of the study are a key factor in the evaluation process. In our study all of them have a strong background knowledge in computer science so they are significantly familiar with the terminology used in the domain. It is also important to highlight that none of them have been informed about the provenance of those terms, so they are not biased towards a more benevolent judgement during the assessment phase therefore ensuring a more objective outcome.

Each term will be evaluated by at least 5 different experts. They will have to react to questions with the form "*Does T belong to the computer graphic domain?*" by answering with a Yes/No response. In order to study how consistent the responses about a term $T$ were, we leverage on the Fleiss' Kappa statistical measure for assessing the reliability of the agreement.

The **Fleiss' Kappa** meassure offers a statistical estimation about the agreement between different evaluators. It is an extension of the Cohens' Kappa [16] method, which is only applicable to two evaluators. The measure is described in Equation 13, where the denominator indicates the degree of agreement between evaluators when the random variables are discarded and the numerator expresses the actual agreement obtained.

$$k = \frac{\overline{p_a} - \overline{p_e}}{1 - \overline{p_e}} \tag{13}$$

In equation 14 the variable $p_a$ is obtained by calculating the mean of the values in $p_i$ (equa-

[25] http://sempub.taln.upf.edu/dricorpus

tion 14) for each term, and the variable $p_e$ is obtained through equation 15.

$$\overline{p_a} = \frac{1}{N}\sum_{i=1}^{N} p_i \tag{14}$$

$$\overline{p_e} = \sum_{j=1}^{k} p_j^2 \tag{15}$$

The variables $p_i$ and $p_j$ used inside equations 14 and 15 are described in equations 16 and 17. The variable $p_j$ (equation 16) expresses the proportion of assignments for each possible answer $j$ (therefore Equation 15 calculates the mean of the squares of those proportions), being $N$ the number of terms considered during the evaluation, $n$ the number of experts that have participated in the evaluation and $\sum_{i=1}^{N} n_{ij}$ the number of experts who have answered with an answer $j$ to the term $i$. Also $p_i$ (equation 17) indicates the ratio of experts that have come to an agreement normalised by the total number of expert pairs possible (the mean of this variable, used in the general equation, is calculated according to equation 14). In the same way than previous equation, $n$ refers to the number of experts having evaluating the proposed concepts, $k$ is the number of valid responses the evaluator can choose, and $\sum_{j=1}^{k} n_{ij}^2$ is the square of the number of assignment available for each term and category.

$$p_j = \frac{1}{Nn}\sum_{i=1}^{N} n_{ij} \tag{16}$$

$$p_i = \frac{1}{n(n-1)}\left[(\textstyle\sum_{j=1}^{k} n_{ij}^2) - (n)\right] \tag{17}$$

The initial value of $k$ will fall inside the interval $[-1,1]$, 1 meaning a full agreement between the evaluators and -1 the complete lack of agreement. If the value of $k$ is lower or very close to 0, the experiment should be reviewed given this is a strong indicator of the lack of coherence in the data for triggering valid solutions.

In figure 4 we can find an example on how to calculate this measure. For this case in particular, the number of terms to be evaluated is $17$, the number of responses is $2$ (yes/no) and the number of evaluators per term is $5$. After executing the different calculations, the final score obtained suggests that the level of agreement between the experts involved in the study is not significant enough to rely on the results indicated by the evaluators' responses. Therefore it would be discouraged to accept those annotations as the basic for an evaluation since the results do not seem conclusive enough.

### 4.4.2 Tools Involved in the *Gold Standard* Creation

In order to perform the experiments working on Dimension 1, we have leveraged on different tools that will allow us to show the results to the experts, and obtain the feedback from them. Data will be available in CSV[26] format, we will rely on common spreadsheets for the calculation

[26] https://es.wikipedia.org/wiki/CSV

| | A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|---|
| 1 | **Term** | **Yes** | **No** | **Pi** | | | |
| 2 | graphics | 5 | 0 | 1 | | | |
| 3 | results | 2 | 3 | 0.4 | | **N** | 17 |
| 4 | images | 5 | 0 | 1 | | **k** | 2 |
| 5 | transactions | 2 | 3 | 0.4 | | **n** | 5 |
| 6 | points | 4 | 1 | 0.6 | | | |
| 7 | models | 4 | 1 | 0.6 | | | |
| 8 | methods | 3 | 2 | 0.4 | | **pa** | 0.5764705882 |
| 9 | data | 3 | 2 | 0.4 | | **pe** | 0.5753633218 |
| 10 | objects | 3 | 2 | 0.4 | | | |
| 11 | values | 2 | 3 | 0.4 | | k | 0.00260756193 |
| 12 | pixel | 5 | 0 | 1 | | | |
| 13 | constraints | 3 | 2 | 0.4 | | | |
| 14 | vertices | 4 | 1 | 0.6 | | | |
| 15 | techniques | 2 | 3 | 0.4 | | | |
| 16 | vertex | 4 | 1 | 0.6 | | | |
| 17 | regions | 4 | 1 | 0.6 | | | |
| 18 | parameters | 4 | 1 | 0.6 | | | |
| 19 | **Pj** | 0.6941176471 | 0.3058823529 | | | | |

Figure 4: Example of obtaining the score

of the agreement scores. However for the display of the candidates to the expert participating in the study and keeping track of the responses we provide, we have leveraged on a specialised Web Site called CrowdFlower[27] that already implements all the necessary interaction and only requires to set the experiment parameters (such as total number of experts, number of responses per term, etc) and the data to be evaluated (in our case the list of terms obtained by our ontology learning approach).

### 4.4.3 Setting up the Experiment with CrowdFlower

CrowdFlower is a tool that allows us to design many kinds of crowdsourcing tasks for labelling data and getting users' feedback, such as data categorisation, sentiment analysis or translation assessment. It manages very heterogeneous data, including images, text, or video. In a nutshell it is a very powerful tool that is currently being exploited by very important international companies.

Below we summarise the main tasks that have been implemented through the Web functionalities offered by CrowdFlower in order to carry on the evaluation and creation of the *Gold Standard*

- **Data Loading:** The first step is to be able to upload the data that will feed the online forms. CrowdFlower makes possible to work with very different file formats (.csv, .tsv, .xls, .xlsx, .ods). In the case of our research and as we introduced before we are going to rely on the CSV format, for specifying a table with the list of terms that need to be assessed.

[27] https://www.crowdflower.com/

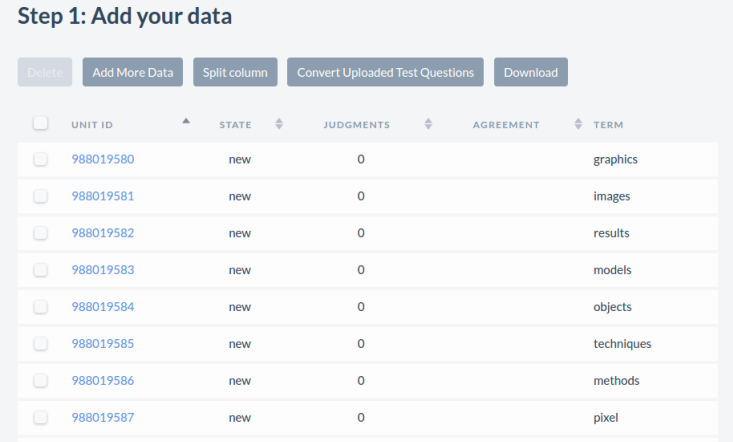


Figure 5: Editing the dataset via the CrowdFlower GUI

– **Dataset Edition:** An intermediate step between the experiment design and the data loading is the edition of the dataset that has just been uploaded to the platform. This step allows to further check on and polish the data being loaded, download the snapshot of the dataset that is available on the server, upload a new dataset from a different file or start creating quality-control questions. In Figure 5 we can observe a screenshot of the aforementioned functionalities.

– **Experiment Design:** It is the most important step inside the generation of the Gold Standard. It involves to decide on the formulation of the question that will be presented to the experts, therefore deeply influencing the kind of feedback they will provide. In our case we have opted for a very simple request: to indicate if the term $T$ belongs or not to the domain of computer graphics. In addition we define the possible answers that can be made (for the current experiment, yes/no), a human readable description with the instructions to fill up the form, and the title of the experiment. Figure 6 shows an screenshot of the CrowdFlower dialog where we can set up those details.

– **Creation of the Assessment Questions:** The next step to be performed is the creation of the assessment questions, whose objective is to check on the level of knowledge the expert has in the matter before providing answers on the real data. As discussed before, if the evaluators are not able to answer those questions properly, the results obtained from their responses will not take into account for the experiments. In order to create the assessment questions we need to select different examples of questions as defined during the previous phase, for which we have already a solid answer for. Representative instances and corner-cases are preferred in order to get more relevant clues on the adequacy of the expert to answer to the questions in the form. For example, we could think about expecting a “no” response to the question about the term “table” belonging to the domain of computer graphics, and“si” for the concept “image”.

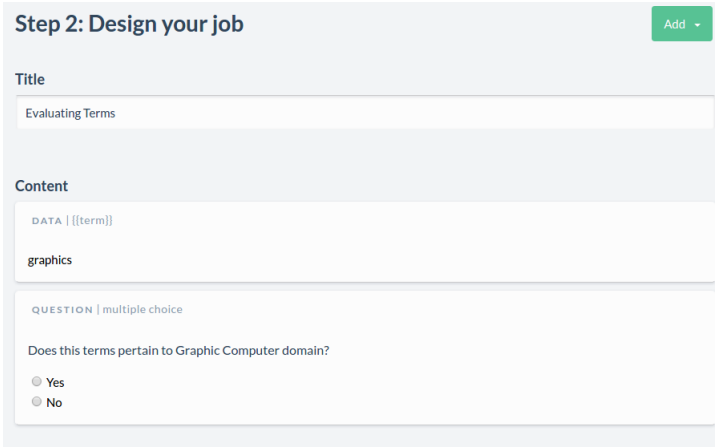


Figure 6: Designing the Experiment via the CrowdFlower GUI

– **Launching the Experiment:** The last step consists on configuring the parameters of the experiments and launching its execution so the experts can start providing answers to the questions. In particular, we need to detail the reward that evaluators will be getting every time a question is answered. Immediately after, we can define if we want only people registered in CrowdFlower to be able to provide responses, or we prefer to generate a URL that we can share with any potential expert that can participate in the experiment. In Figure 7 we can observe the online form that evaluators access in order to read the questions and provide answers to them.

#### 4.4.4 Results for Dimension 1

In this subsection we detail and analyse the feedback obtained from the experts who participated in the Gold Standard generation, corresponding to the First Dimension in the proposed methodology (see Section 4.2.2), together with the agreement scores obtained according to Fleiss' Kappa. The datasets and all the calculations made in order to obtained the scores we will report below can be found at the following URL: `https://figshare.com/articles/Evaluation_Data_Set_xlsx/3485690/2`.

The learning process applied over the SIGGRAPH corpus produced a set of 1788 nouns that were potentially relevant to the domain. Each of those terms have a relevance score associated to them, also called *termhood*. This score is a statistical measure for judging on the relevance of a term according to the corpus where it has been extracted from, and it has been described at [39]. Starting from the long list of learned terms, we decided to filter out the least relevant results by discarding the terms with a *termhood* lower than $0.24$, in order to end up with a set of 104 terms that were potentially more relevant and affordable to be assessed by the experts in a reasonable amount of time. A total of 17 evaluators, all of them familiar with the field of "*Computer Science*" answered to 520 different questions, 5 per each term, and

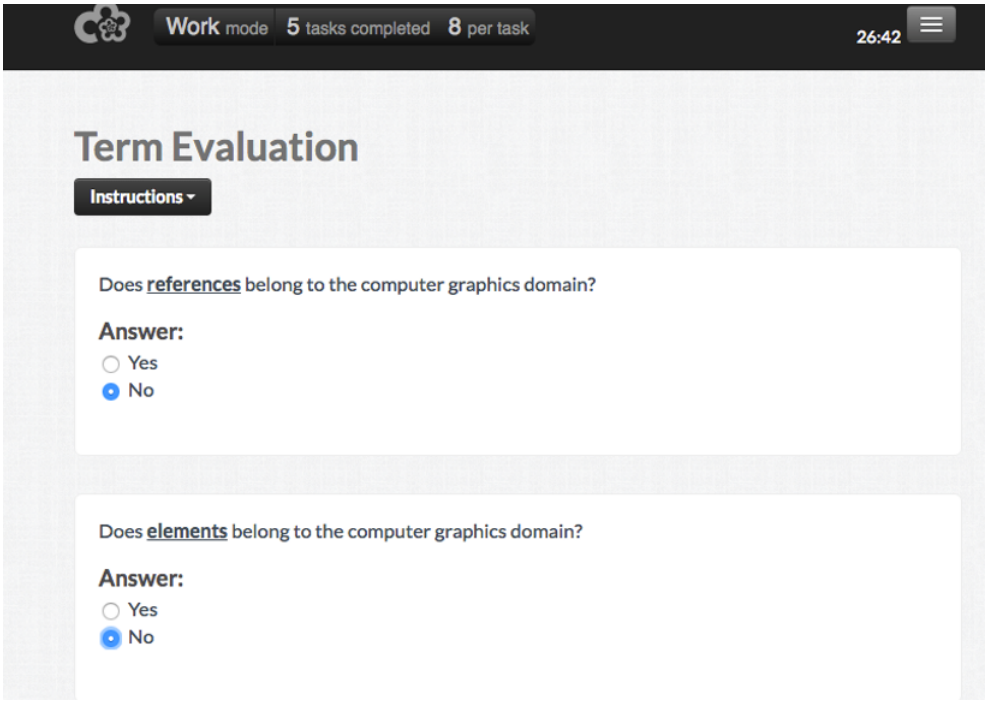


Figure 7: Screenshot of the Assessment form in CrowdFlower

| Termhood | Precision | Recall | F-Measure |
|---|---|---|---|
| **0.23** | 67.3% | 100% | 80.45% |
| **0.27** | 68.7% | 47.1% | 55.9% |
| **0.32** | 81.8% | 25.7% | 39.1% |

Table 1: Evaluation Mesures for Dimension 1 (Lexical) when varying the *termhood* used as threshold

12 quality control questions. In Table 1 we summarise the obtained results in terms of Recall, Precision and F-measures (having $\beta = 1$). In addition in Table 2 we detail the Precision for the first 5, 10, 20, 20 and 40 relevant terms in order to check how this measure evolves when we consider a bigger set of candidate terms.

If we take a closer look to the results in Table 1 we can observe how the higher the value we assign to the *termhood* threshold, the higher the precision is, while Recall and F-Measure decrease. This is indicating that by tightening the threshold we are able to increase the proportion of valid terms that the resulting subset is containing. At the same time, the ratio of valid

| Termhood | Precision5 | Precision10 | Precision20 | Precision30 | Precision40 |
|---|---|---|---|---|---|
| **0.23** | 60% | 70% | 80% | 80% | 75% |

Table 2: Precision for Dimension 1 (Lexical) at 5, 10, 20, 30 and 40

| | Normal | Black List | Agreement Filtered |
|---|---|---|---|
| **k** | 0.16 | 0.18 | 0.38 |

Table 3: Level of agreement in the Experts' responses

terms compared to the total amount of terms obtained from the entire set of submitted terms is decreased.

On the other hand, if we analyse the figures reported in Table 2 we can conclude that the precision increases when we consider up to 20 top terms ordered according to the *termhood*, while if we keep extending the set to lower positions (30, 40 or more) the precision starts to fall to less competitive scores. This indicates that after position 20 the number of relevant terms we find is much lower compared with the number of unimportant terms that are not related to computer graphics and therefore finally discarded by the experts.

**Level of Agreement among Experts**

As we introduced before, the level of agreement among experts have been calculated according tho the Fleiss' Kappa method. Table 3 reflects how given the initial agreement score was quite low, we opted for implementing two different variations in order to obtain a more adequate score to rely on.

Concerning the first approach, we have selected a *black list* of terms that are very related with science in general and therefore are very present in the documents where the ontology learning algorithm has been applied over, but they are irrelevant for a particular domain since they are assumed to be known. Examples of such terms are: *results, values, etc*. By applying this filtering over the original 104 terms considered in the evaluation, we end up having a reduced set of 76. The other approach, which we have called (*Agreement Filtered*), we decided to directly discard all the terms with a $p_i$ higher or equal to $0.6$. This way we discard beforehand all the terms where the expert struggled to give a clear judgement on, producing a smaller subset of 60 terms. Even the agreement score increases with this last approach, the precision is reduced significantly. This indicates that terms that were removed were for the most part relevant terms, but the experts had troubles in correctly classifying them and therefore find a trustable agreement. This is suggesting that a more precise design of the experiment is needed for future efforts in this line, probably by better formulating the question and providing better guidelines and examples that do not cause so much confusion between the experts when providing their responses. We plan to study the effect of such adjustments in future developments.

### 4.4.5 Evaluation of Relations

The ontology learning algorithm described in Section 3 does not only produce terms but also identifies relations between them as explained in Section 3.2. As we previously stated, only hypernymy relations are supported at the moment.

The Dimension 2 $(\omega_R, F_{Hybrid})$ considered in the $M_{OntoLearn}$ methodology does consider the evaluation of relations between the different individuals in the ontology. However, a first manual study of the results obtained using the current version of the learning algorithm reveals that the results are significantly lacking the minimum quality that is needed to start putting efforts towards a formal evaluation strategy of this dimension. In particular a lightweight assessment of the top 100 relations found by this method is reporting a precision around 0.16, which is clearly low to be acceptable.

However, this preliminary study has also triggered some interesting conclusions that are worth to be reported and make us remain optimistic about the future possibilities of the implemented algorithm:

- Some of the Relations are well spotted and relevant to the domain. For example the hypernymy between the words **[optical** $\rightarrow$ **camera]** or between **[format** $\rightarrow$ **digital]** is well spotted and reveals that the algorithm is successful in some cases.

- Some of the Relations are well spotted and belong to the scientific domain, but they are applicable to other research areas. For example, there is a valid hypernymy in the pairs of words *[sequential* $\rightarrow$ *decision]* and **[future** $\rightarrow$ **work]**, but they be may be present in any science-related corpus.

- Some of the relations are well spotted, but they are irrelevant to the domain being studied. For example, **[objects** $\rightarrow$ **fragment]** are simply out of the scope of the field that we are focusing over.

- Some relations express some degree of semantic connection, but not exactly hypernymy. For example paris like **[complex** $\rightarrow$ **simplicial]** are indicating antonymy, and others such as **[work** $\rightarrow$ **researcher]** are somehow related but the connection can not be classified inside any of the well-know semantic relations, such as meronymy, antonymy, hypernymy or synonymy.

We expect that by being able to spot those particular situations we can implement more advanced versions of the relation extraction algorithm, hence increasing the precision of this tool for finding relations between terms and better justifying the need of a formal evaluation of the Dimension 2 according to the proposed methodology $M_{OntoLearn}$.

# 5 Conclusions and Future Work

The complexity of today's scientific ecosystems justifies the need of techniques that automatically unveil the most relevant concepts and relations that are represented in big corpora of research objects. This way we can be able to understand the kind of knowledge that is available inside them without the burden of manually interpreting every single document. Hence in DRInventor we have developed an ontology learning algorithm order to spot relevant terms and relations that can be later leveraged on.

The contributions of the implemented ontology learning approach are twofold. On the one hand, the term extraction algorithm relies on different statistical methods aiming to discover how relevant a term is according to the corpora where it has been extracted from. In particular, measures such as Domain Consensus, Domain Relevance, or C-value/NC have been linearly combined to provide a final score that allows us filtering out irrelevant terms from the set of concepts describing the corpora. On the other hand, we have tackled the difficulty for finding labelled data about relations between terms, what makes itvery difficult to implement supervised algorithms that learn them automatically. In particular we have developed a distant supervision approach that can be trained over a general dataset instead (in our case, Wikipedia).

In addition, we have presented a methodology ($M_{OntoLearn}$) to evaluate the result of ontology learning approaches over big unstructured corpora. Starting from the definition of a *flatten ontology* as a simplified formalisation that better matches the current trends in information extraction and ontology learning where more specific and changeable domains are involved, we have defined a multidimensional methodology that distributes into three well identified levels of complexity $d_W, d_R, d_P$ the different evaluation objetives that are involved in the learning process: lexical, relational, and global aspects. The methodology uses an innovative evaluation method labeled as $F_{Hybrid}$, which takes advantage of the reproducibility of corpus-based solutions, while minimising the cost and promoting a higher recall and precision during the annotation phase. Through an analysis of previously published efforts on ontology learning systems and their evaluation, we have shown how they fall into some of the evaluation dimensions that we identified previously, and how they could further benefit from applying our methodology' guidelines for a more standardised, less-costly way of targeting the crucial evaluation process.

Preliminary evaluation efforts of the results of our ontology learning approach applied over DRInventor corpus have produced some still immature but promising results: the term extraction algorithm with a simple linear combination of the term-hood dimensions has obtained a Precision score of 67.3%. The logic for spotting relations, which intrinsically is difficult to implement given the cognitive complexity of the task and its ambiguity, has obtained a much lower precision according to some preliminary and lightweight assessments. However a deeper study on particular examples suggests that the learning process can be further refined in order to be able to discard those very general relations that, despite being valid from a formal point

of view, are not relevant to the domain.

In the future, we will keep tackling the problem of ontology learning over big corpora of research objects, specifically focusing on the following aspects:

- At the moment, the measures considered to spot relevant terms are combined linearly. Instead of relying in this very simple mechanism to integrate them, we can consider them as features that together with other contextual clues (such as the kind of domain the corpus is addressing, the main topics involved, etc) can feed a classifier that more precisely discovers in which degree each of them should be considered for generating the final relevance score.
- The distant supervision implemented for discovering relations between terms is based on a very general corpora and targeting very general semantic relations. We want to study whether or not relying on more specific corpus in the scientific domain can significantly improve the quality of the obtained results.
- In the current implementation of the algorithm spotting relations, only connections between terms inside the same sentence are considered. In future developments, and even the distance between the candidate words is still a key feature to decide on the strength of the relation between them, we are interested in considering also connections between terms in different but proximal sentences inside the text.
- The finer grained relation types that can be obtained through the application of ideas described in previous point can help to better discard non relevant connections, as reported in our initial findings at Section 4.4.5. This more accurate selection can help to increase the precision of the current method.
- For the evaluation methodology $M_{OntoLearn}$, we want to keep moving towards a less human-based intervention evaluation methods, mainly by researching on the level of automation and accuracy of the results produced by state-of-the-art ontology learning algorithms using synthetic measures.
- Concerning the evaluation of Dimension 1 that has been already performed, we are aiming for an update in the experimental protocol in order to consider $F_{Hybrid}$ methods instead of a simple assessments of experts. Also, following the efforts evaluating Terms (Dimension 1 in $M_{OntoLearn}$), we plan to develop Gold Standards addressing other ontology aspects, such as relations, restrictions, rules, etc.
- Based on the previous points and considering that soon we will have available a valid set of labelled data able to support a more complete evaluation, it would be very interesting to test the performance of DRInventor ontology learning methods against other approaches in the literature so as to decide on their adequacy for particular tasks.

## References


[1] Eneko Agirre, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. A study on similarity and relatedness using distributional and wordnet-based approaches. In *The 2009 Conference of the North American Chapter of the Association for Computational Linguistics*, pages 19–27, 2009.

[2] Harith Alani, Sanghee Kim, David E Millard, Mark J Weal, Wendy Hall, Paul H Lewis, and Nigel R Shadbolt. Automatic ontology-based knowledge extraction from web documents. *IEEE Intelligent Systems*, 18(1):14–21, 2003.

[3] Enrique Alfonseca, Diana Pérez, and Pilar Rodríguez. Welkin: automatic generation of adaptive hypermedia sites with nlp techniques. In *International Conference on Web Engineering*, pages 617–618. Springer, 2004.

[4] Carlos Badenes, Rafael Gonzalez, Oscar Corcho, and Feng Dong. Repository of indexed ros. Technical report, 2015.

[5] Chris Biemann. Ontology learning from text: A survey of methods. In *LDV forum*, volume 20, pages 75–93, 2005.

[6] Georgeta Bordea, Paul Buitelaar, Stefano Faralli, and Roberto Navigli. Semeval-2015 task 17: Taxonomy extraction evaluation (texeval). *SemEval-2015*, 452(465), 2015.

[7] Razvan C. Bunescu and Raymond J. Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 724–731, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[8] Steven L Camiña. *A comparison of taxonomy generation techniques using bibliometric methods: applied to research strategy formulation*. PhD thesis, Massachusetts Institute of Technology, 2010.

[9] Philipp Cimiano and Johanna Völker. text2onto. In *International Conference on Application of Natural Language to Information Systems*, pages 227–238, 2005.

[10] W Bruce Croft, Donald Metzler, and Trevor Strohman. *Search engines: Information retrieval in practice*. Addison-Wesley Reading, 2010.

[11] Klaas Dellschaft and Steffen Staab. On how to perform a gold standard based evaluation of ontology learning. In *International Semantic Web Conference*, pages 228–241. Springer, 2006.

[12] Klaas Dellschaft and Steffen Staab. Strategies for the evaluation of ontology learning. In *Ontology Learning and Population*, volume 167, pages 253–272, 2008.

[13] Marc Ehrig, Peter Haase, Mark Hefke, and Nenad Stojanovic. Similarity for ontologies-a comprehensive framework. *ECIS 2005 Proceedings*, page 127, 2005.

[14] Blaz Fortuna, Marko Grobelnik, and Dunja Mladenic. *OntoGen: Semi-automatic Ontology Editor*, pages 309–318. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[15] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130, 2000.

[16] Francis Galton. *Finger prints*. Macmillan and Company, 1892.

[17] Xinbo Gao, Bing Xiao, Dacheng Tao, and Xuelong Li. A survey of graph edit distance. *Pattern Analysis and applications*, 13(1):113–129, 2010.

[18] José Luis Redondo García, Giuseppe Rizzo, and Raphaël Troncy. The Concentric Nature of News Semantic Snapshots. In *$8^{th}$ international Conference on Knowledge Capture (KCAP)*, 2015.

[19] Asunción Gómez-Peréz and David Manzano-Macho. An overview of methods and tools for ontology learning from texts. *The Knowledge Engineering Review*, 19(3):187âĂŞ212, Sep 2004.

[20] Nicola Guarino and Christopher A Welty. An overview of ontoclean. In *Handbook on ontologies*, pages 201–220. Springer, 2009.

[21] Udo Hahn and Martin Romacker. The syndikate text knowledge base generator. In *1st international conference on Human language technology research*, pages 1–6, 2001.

[22] Xing Jiang and Ah-Hwee Tan. Crctol: A semantic-based domain ontology learning system. *Journal of the American Society for Information Science and Technology*, 61(1):150–168, 2010.

[23] José Paulo Leal, Vânia Rodrigues, and Ricardo Queirós. Computing semantic relatedness using dbpedia. In *OASIcs-OpenAccess Series in Informatics*, volume 21, 2012.

[24] David Manzano, Asunción Gómez-Pérez, and Daniel Borrajo. Unsupervised and domain independent ontology learning: combining heterogeneous sources of evidence. 2008.

[25] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language*

*Processing of the AFNLP: Volume 2 - Volume 2*, ACL '09, pages 1003–1011. Association for Computational Linguistics, 2009.

[26] Joel Palmius. Criteria for measuring and comparing information systems. 2007.

[27] Robert Porzel and Rainer Malaka. A task-based approach for ontology evaluation. In *ECAI Workshop on Ontology Learning and Population, Valencia, Spain*. Citeseer, 2004.

[28] J Ramanand, Akshay Ukey, Brahm Kiran Singh, and Pushpak Bhattacharyya. Mapping and structural analysis of multi-lingual wordnets. *IEEE Data Eng. Bull.*, 30(1):30–43, 2007.

[29] José Luis Redondo García, Carlos Badenes, and Oscar Corcho. Techniques of personalised recommendations of ros with reports. Technical report, 2016.

[30] Marta Sabou, Chris Wroe, Carole Goble, and Gilad Mishne. Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *14th international conference on World Wide Web*, pages 190–198. ACM, 2005.

[31] F. Sclano and P. Velardi. *TermExtractor: a Web Application to Learn the Shared Terminology of Emergent Web Communities*, pages 287–290. Springer London, London, 2007.

[32] Mehrnoush Shamsfard and Ahmad Barforoush. An introduction to hasti: An ontology learning system. In *Proceedings of the iasted international conference artificial intelligence and soft computing, Acta Press, Galgary, Canada*, pages 242–247, 2002.

[33] Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. Learning syntactic patterns for automatic hypernym discovery. In *Advances in Neural Information Processing Systems (NIPS 2004)*, November 2004. This is a draft version from the NIPS preproceedings; the final version will be published by April 2005.

[34] Hristo Tanev and Bernardo Magnini. Weakly supervised approaches for ontology population. In *Conference on Ontology Learning and Population*, pages 129–143, 2008.

[35] Paola Velardi, Roberto Navigli, Alessandro Cuchiarelli, and R Neri. Evaluation of ontolearn, a methodology for automatic learning of domain ontologies. *Ontology Learning from Text: Methods, evaluation and applications*, 123:92, 2005.

[36] Wilson Wong, Wei Liu, and Mohammed Bennamoun. Ontology learning from text: A look back and into the future. *ACM Computing Surveys (CSUR)*, 44(4):20, 2012.

[37] Laura A Zager and George C Verghese. Graph similarity scoring and matching. *Applied mathematics letters*, 21(1):86–94, 2008.

[38] Elias Zavitsanos, George Paliouras, and George A Vouros. Gold standard evaluation of ontology learning methods through ontology transformation and alignment. *IEEE Transactions on Knowledge and Data Engineering*, 23(11):1635–1648, 2011.

 

[39] Ziqi Zhang, José Iria, Christopher Brewster, and Fabio Ciravegna. A comparative evaluation of term recognition algorithms. In *LREC*, 2008.