Ontology Learning from Open Linked Data and Web Snippets

Ilaria Tiddi, Nesrine Ben Mustapha, Yves Vanrompay, and Marie-Aude Aufaure

Ecole Centrale Paris, France
{ilaria.tiddi,nesrine.ben-mustapha,yves.vanrompay, marie-aude.aufaure}@ecp.fr

Abstract. The Web of Open Linked Data (OLD) is a recommended best practice for exposing, sharing, and connecting pieces of data, information, and knowledge on the Semantic Web using URIs and RDF. Such data can be used as a training source for ontology learning from web textual contents in order to bridge the gap between structured data and the Web. In this paper, we propose a new method of ontology learning that consists in learning linguistic patterns related to OLD entities attributes from web snippets. Our insight is to use the Linked Data as a skeleton for ontology construction and for pattern learning from texts. The contribution resides on learning patterns for relations existing in the Web of Linked Data from Web content. These patterns are used to populate the ontology core schema with new entities and attributes values. The experiments of the proposal have shown promising results in precision.

1 Introduction

The Web of Linked Data contains solid structured data and aims to link them from different sources using equivalence statements based on an ontological representation. Efforts made by the Semantic Web community to connect data from diverse domains ensure their reliability and disambiguation. Moreover, the Web semantic evolution, by linking and structuring data, avoids the long manual ontology building process, and ontology learning techniques can take advantage from this. Such approaches aim at building ontologies from knowledge sources using a set of machine learning techniques and knowledge acquisition methods. Text-mining techniques for enriching ontologies with concepts and relationships starting from texts have been widely used in the knowledge engineering community. More recently, with the growth of the Web, it has become important to exploit its unstructured textual contents for knowledge acquisition.

In addition to this, a huge amount of information is available as a Web of structured data, in form of shared and open knowledge. Many efforts have been made so far in view of a global adoption of Linked Data[1]. Searching over aggregated data, semantic querying, and applications operating over global data

[1] http://linkeddata.org/

P. Herrero et al. (Eds.): OTM 2012 Workshops, LNCS 7567, pp. 434–443, 2012.

are just some of the main objectives for this new data sources format appearing in the Web[1]. Despite the Semantic Web Community's efforts to provide techniques for linking entities between knowledge sources, it is necessary to bridge the gap between structured data of the Semantic Web and web textual contents. Hence, ontology learning can tackle this issue by providing techniques able to structure a large amount of unstructured data without any human intervention.

In this paper, our first aim is to exploit these linked data as a starting point for ontology construction. The Linked Data is increasing in size and connections, and it can be useful to exploit the richness and scalability of this knowledge. Our insight is to use the **Linked Data as a skeleton for ontology construction and for pattern learning from texts**. The novelty is to focus on learning patterns for relations existing in the Web of Linked Data from Web snippets. These patterns are used to populate the ontology core schema with new entities and attributes values. In order to do this, we propose a hybrid approach based on linguistic and statistical techniques for **Pattern-based Entity Discovery from Web Snippets for ontology population**.

This paper is organized as follows. Section 2 presents related works. In Section 3 and 4, we present our approach followed by some experiments we conducted. In the final section, we discuss conclusions and future work.

2 Related Work

Linguistic patterns have been used in many fields namely in non-supervised information extraction and ontology learning. In linguistic approaches, lexico-syntactic patterns are manually defined by linguists and used for taxonomic relation discovery. Hearst pattern-based techniques have been successfully scaled up to the Web in order to identify certain types of relations such as hyponymy or meronymy[3]. *Lexico-syntactic patterns* are constructions that can indicate an interesting relation. Those satisfy the following needs: (i) they occur frequently, (ii) they (almost) always indicate the relation of interest and (iii) they can be recognized with little or no pre-encoded knowledge[4]. Some recent approaches have been proposed to automatically discover patterns. In [11], [5], [12], [7], authors focus on approaches using existing ontology to extract concepts linked by a relationship, and produce lexico-syntactic patterns. The interaction between natural language patterns and ontology relations for ontology population or entities extraction has been already explored.

However, with the increasing need for automatic pattern identification, research has focused on the use of *dependency grammars* for Named Entities Recognition and relation extraction (as an extension of it). *Dependency Trees* represent the syntactic relations between words by a list of tuples in the form *grammarRelation(regentWord, dependentWord)*. Each node is a word with its syntactic label, and each edge a grammatical relation between two words. Dependency patterns are the shortest path linking two words, the instance and its attribute value. This representation also includes the semantic information of a sentence, favoring the extraction relation linking two words[2]. Many efforts have been made using dependency parsing for semantic relations discovery

or entity recognition. In [8], [9], [6], relations are extracted using dependency parsing. These approaches are mainly focused on specific grammatical relations, corresponding to the most common ones.

New trends focus on the use of Linked Data for data-mining and ontology matching [3],[13]. However, for the best of our knowledge, they have not been exploited yet for dependencies-based ontology learning. The main challenge of the present work is then to use structured data as training corpora, for dependency structures discovery from unstructured textual contents of the Web.

3 Pattern-Based Ontology Construction

We present our novel approach for ontology learning, using the Linked Data and Web snippets for pattern learning and entity discovery. Our structured approach consists in five main steps, which can be seen in Figure 1.

(1) `Linked Data Extraction`. Focusing on the DBPedia and the Schema.org datasets, which can be considered the backbone of the Linked Data, an existing concept (i.e. selected from the user) is retrieved with its attributes, the instances and their attributes values. Equivalence statements such as "same-as" are used to navigate the web of Linked Data, in order to discover new relations and update the attribute set.
(2) `Web Search`. Considering a couple *instance-attribute value*, the corpus constitution is led using web snippets provided by a search engine, instead of noisy Web pages.

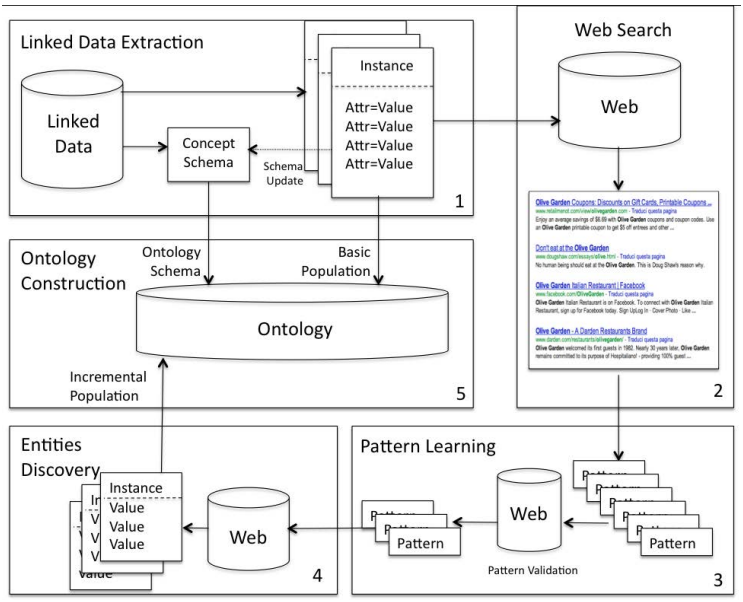

Fig. 1. General Architecture

(3) `Pattern Learning`. Snippets are parsed using Dependency Grammars and associated to a tree. The dependency trees are the key for the patterns collection and validatation. Patterns are discovered by extracting relations between entities.
(4) `Entity Discovery`. Validated dependency patterns are used to query the Web in order to discover new attribute values from new web snippets.
(5) `Ontology Population`. Extracted instances and their attribute values are used for the ontology enrichment.

In the next subsections, the different steps of the proposed approach will be detailed.

3.1 Linked Data Extraction

In order to construct the training corpus for patterns learning, the ontology schema with some related instances are extracted from the linked data knowledge. This step is composed of two tasks, as explained below.

The *Core Schema Extraction* aims to obtain a schema for a concept in the basic ontology provided, for instance, by the user. A set of attributes and some instances related to it alre also included in the schema. The concept and a basic set of attributes is extracted from the Schema.org ontology[2], which has been introduced by the main search engines in order to help web masters to semantically annotate their sites. This ontology provides standardized attributes for several domain concepts, avoiding redundancy or ambiguity in data structures. We then use the Schema.org structure as a starting point for the ontology schema construction.

The *Entities and Attributes values extraction* uses the DBPedia dataset[3] for instances and attributes values extraction. DBPedia is the backbone of the Open Linked Data. Thank's to its interlinking efforts, DBPedia is nowadays the intersection point among the huge Semantic Web Dataset and includes ontologies such as YAGO2, MusicBrainz, GeoNames, WordNet and many others. We exploit this interconnections for retrieving entities and attributes values of a same concept. As certain connections in the web of Linked Data may still be missing, we provide with a matching between Schema.org and DBPedia data, in order to avoid entities ambiguity.

The general schema of the ontology is a combination from the matchings between the *Schema.org* extracted schema and DBPedia instances.

3.2 Web Queries

This step consists in building a training corpus from the Web for the patterns learning. For example, considering a triple

`<OliveGarden><parentCompany><DardenRestaurants>`

web snippets containing the instance and its attribute value are extracted, by building a web query such as *"Olive Garden*Darden Restaurants"*. The term

[2] `http://schema.rdfs.org/`
[3] `http://dbpedia.org/`

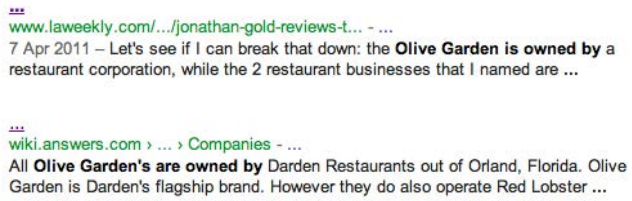

Fig. 2. Web Snippets examples

"snippet" we use here denotes a fragment of a Web page returned by remote search engines (such as Google or Yahoo!) and summarizing the context of searched pairs, as shown in Figure 2.

Our assomption, widely demonstrated by the literature, is that for each relation there exist in the natural language some universal grammatical patterns for it. The hypothesis is that the use of Dependency Grammars for the patterns design can be more efficient for the in information extraction task.

The set of resulting snippets composes the training corpus for the pattern learning.

3.3 Pattern Learning

The objective of this step is to extract a set of candidate lexico-syntactic patterns for a specific attribute. For instance, patterns for the attribute <parentCompany> may be: "X_{NP}, ***operates under*** Y_{NP}", "***Owned by*** Y_{NP}, X_{NP} ..." or "X_{NP}, ***part of*** Y_{NP}" For this purpose, Natural Language Processing techniques are explored here. The step is composed by two phases: the extraction of a set of candidate patterns from the training corpus, and their ranking by using their frequencies.

The main objective of the *Dependency Analysis* phase is the acquisition of semantic relations expressed in natural language texts. Considering an input pair (the instance and its attribute value) and a sentence from the Web corpus, the structure of the latter is explored in order to retrieve the shortest *dependency path* between the pair values, as shown in Algorithm 1. Therefore, sentences are analyzed through a dependency grammar parser[4]. Candidate patterns collected are validated in the *Pattern Ranking* phase. In order to evaluate their accuracy, we group patterns according to their dependencies. For instance, the pattern $nsubjpass([start|find], X), agent([start|find], Y)$, refers to a dependency tree including a passive nominal subject relation and an agent relation, linking the X and the Y words with the verb is *start* or *find*. The selection of the best candidates is based on the distributional analysis of patterns in the training corpus.

For validation purposes, we adapted the well known measure of *Term Frequency/Inverse Document Frequency* ($tf * idf$) to our corpus and filter patterns using their frequencies. The candidate pattern frequencies (i.e., how many times the pattern is used in a single attribute), are multiplied with their inverse frequencies in each attribute (i.e., the general importance of such pattern among a

[4] `http://nlp.stanford.edu/software/lex-parser.shtml`

Algorithm 1. Pattern Extraction for an attribute

```
corpus ← Entity1 + Entity2 : Sentence
patternsForAttribute ← emptySet()
for i = 0 to size(corpus) do
    dependencyTree ← parseSentence(Sentence);
    head1 ← getHead(Entity1);
    head2 ← getHead(Entity2);
    pattern ← getPathBetweenNodes(head1, head2, dependencyTree);
    if pattern exists then
        patternsForAttribute ← patternsForAttribute + pattern;
    end if
end for
```

set of attributes) to obtain the candidate pattern importance. With this *Pattern Frequency/Inverse Document Frequency* ($pf * idf$) approach, most frequent patterns are discarded only if their frequency is too high in all the attributes set. The measure is computed as follows:

$$pf * idf(p, a, A) = pf(p, a) \times log \frac{|A|}{|\{a \in A : p \in a\}|} \quad (1)$$

where p is the candidate pattern, A the number of existing attributes and $|\{a \in A : p \in a\}|$ the number of attributes where the pattern p appears.

3.4 Entity Discovery

This process aims to extract new candidate entities from snippets provided by the search engine. Web queries are formulated using the validated patterns and a test set of entities, as follows: "CANDIDATE **is owned by*". Given the dependency tree of the resulting snippet, and the pattern, a matching between them is executed with the purpose of retrieving node entities. Considering a snippet sentence as "*Pizza Express is owned by One World Enterprises*", the resulting nodes will be `X=Express` and `Y=Enterprises`. A further step is necessary in order to retrieve the whole entity, in case it has children-dependencies ($\texttt{Child}_X$`=Pizza` and $\texttt{Child}_Y$`=One,World`). Figure 3 shows an example of the tree-matching phase.

3.5 Ontology Construction

The final phase concerns the ontology population with new discovered entities and attribute values. The input queries are made up with the validated patterns, and the ontology concept (i.e. *"Restaurant is owned by"*). While new entities (in our case, *Pizza Express*), corresponding to the X part of the pattern, are instance of the concept (*Restaurant*), attributes values are extracted with the Y part of the pattern (*One World Enterprises*).

This steps is shown in Figure 3.

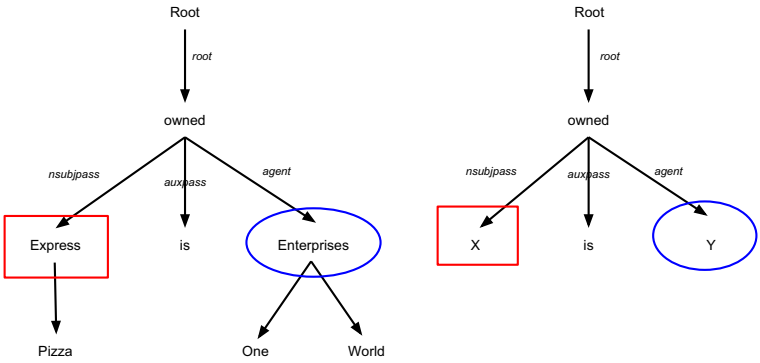

Fig. 3. Tree matching

4 Experimental Evaluation

In this section we give an overview of experiments aiming to evaluate our proposal. We are particularly interested in deriving quantitative insights about the accuracy of our pattern-based extraction approach.

Datasets and Corpora. The evaluation uses three datasets, partitioned into a training and a test corpus. Besides, we used the Schema.org ontology and DBPedia as structured knowledge bases, and the Web as unstructured textual corpus.

Pattern learning phase is carried on using the *Training corpus* built on these two types of sources. We used the Jena framework[5], and the SPARQL language for treating and querying data.

- From the **Schema.org ontology** we extract the basic structure for a given concept. It contains a fixed number of attributes we use for the core schema building of the ontology. We ran our experiment on the *restaurant* main concept.
- From the **DBPedia dataset** 40 instances and attributes values are selected, in order to build the web queries for the search engine.
- 20 snippets for each instance are extracted from the **Web** by using the Bing! Search API[6]. The resulting corpus contains almost 10000 snippets to parse.

A *Test corpus* has been built in order to apply the patterns extraction within the Entities Discovery step. Considering the main concept *Restaurant*, we focused our attention on 5 relations to build the test corpus: *location, founder, keyPerson, parentCompany, products.*

- 100 DBPedia new entities are extracted in order to reformulate new queries.
- 5 snippets for each entity are extracted from the Web and parsed for attribute values extraction.

[5] `http://jena.apache.org/`
[6] `http://www.bing.com/toolbox/bingdeveloper/`

Evaluation Criteria. Evaluation is based on two criteria.

Precision specifies whether attribute values are correctly extracted. It measures the percentage of correctly selected entities (attribute values) in relation to the total number of selected values.

$$Precision = \frac{correctly\_selected\_values}{total\_extracted\_values} \tag{2}$$

Recall shows how much of the existing knowledge is extracted. It is defined as follows:

$$Recall = \frac{correctly\_selected\_values}{total\_correct\_values} \tag{3}$$

We use DBPedia to judge the correctness of extracted values. If the DBPedia dataset contains the value and is related to the entity with the right attribute relation, it is considered correct. For instance, if the system retrieves *Y=Goldola Holdings* and the entity *X=Pizza Express* has an attribute *parentCompany* with `Gondola Holdings` as value, the discovered value is correct.
F-Measure is then calculated as:

$$F - Measure = \frac{2(Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

Results. The $pf * idf$ ranking helps to filter on patterns according to their distribution. General common patterns, such as $nn[X, Y]$ (*"Darden Olive Garden"*) or $poss[Y, X]$ (*"Darden's Olive Garden"*), need to be discarded because of their low precision. On one side, it would be difficult to automatically decide to which attribute the extracted entities are related to (i.e. is *Darden* a person or an organization?). On the other, common patterns have a too large coverage on values retrieval and increase cases of syntactic ambiguity, particularly with complex noun phrases.

We evaluated the correctness of the extracted patterns for each attribute.

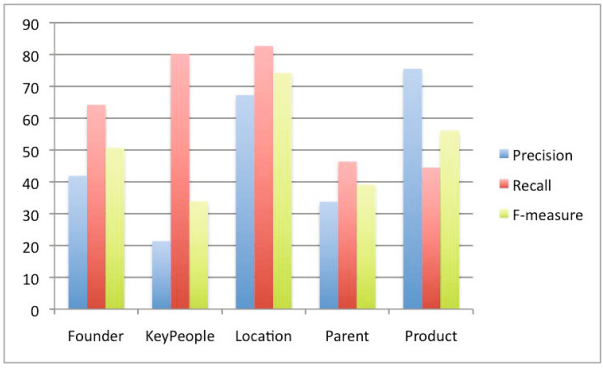

Fig. 4. Histogram for Precision, Recall, F-Measure

In general, better results are produced on attributes more frequent and widespread, as *location*, *founder* or *product*. Low scores are mainly due to the general evaluation of each attribute. In fact, by analyzing in detail each pattern result, an outstanding difference exists between patterns producing a high precision and recall scores, and the ones giving worse results. Hence, we show a summary table for the attribute *parentCompany*.

Summary table for ParentCompany				
Pattern	Freq.	P	R	F
$nsubjpass(owned, X), agent(owned, Y)$ *Ex. "X is owned by Y"*	47	95,7%	63.4%	76.3%
$partmod(X, owned), agent(owned, Y)$ *Ex. "X, owned by Y"; "X, which is owned by Y"*	19	84.2%	10.7%	19.1%
$appos(X, *), prep\_of(*, Y)$ $* =$ (subsidiary|part) *Ex. "X, subsidiary of Y"; "X, part of Y"*	5	100%	13.1%	23.2%
$nsubj(operates, X), prep\_under(operates, Y)$ *Ex. "X operates under Y"*	3	66.6%	66.6%	66.6%

Promising results are obtained in precision. An higher precision is attempted by patterns containing a verbal node. The use of grammar relations reveals a good choice since we are able to retrieve sentences with different syntactic structures with a same pattern (such as "*Pizza Express, owned by One World Enterprises*", "Pizza Express, which is owned by One World Enterprises"). However, some promising patterns have a too low frequency to be considered significant. Low recall is also due to external factors. First, results from web snippets can be non linguistically plain sentences: cut sentences, single linguistic phrases and other kind of noise can be returned from the search engines, and additional steps are necessary to avoid these phenomena. Second, short sentences of snippets may generate errors in parsing and affect precision and recall scores. Finally, using a bigger number of instances in the Entity Discovery phase is certainly necessary in order to increase recall in the approach.

5 Conclusion and Future Work

In summary, the idea behind this work is to combine unstructured and structured data using linguistic and machine learning techniques, with the purpose of the enrichment of an ontology using web snippets. The structure given from the Linked Data provide a good basis and initial results are promising. The dependency patterns we learn return a high precision, but some work is still necessary for improving them. We have put forward some future directions in which this might be done.

We intend to investigate the dynamic enrichment of Open Linked Data from unstructured web contents. The use of Linked Data knowledge could be promising also for entities classification or as test corpus, instead of web snippets. One

further possibility is the treatment of snippets, or any other short textual content. Coreference resolution is also a step to explore for increasing parsing results. Unstructured information is often represented as multiple sentences referring to the same entity and linked by anaphoras or cataphoras. Then, coreference resolution would be a good way to increase accuracy of dependency patterns. On the other side, we are aware that we need to investigate for a bigger test corpus easier to parse and analyze, in order to avoid the affection of our pattenr learning.

References

1. Bizer, C., Heath, T., Berners-Lee, T.: Linked Data - The Story So Far. Int. J. Semantic Web Inf. Syst. 5(3), 1–22 (2009)
2. Marneffe, M.C., Manning, C.D.: The Stanford typed dependencies representation. In: Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation (CrossParser 2008), pp. 1–8. Association for Computational Linguistics, Stroudsburg (2008)
3. Sabou, M., Fernandez, M., Motta, E.: Evaluating Semantic Relations by Exploring Ontologies on the Semantic Web. In: Horacek, H., Métais, E., Muñoz, R., Wolska, M. (eds.) NLDB 2009. LNCS, vol. 5723, pp. 269–280. Springer, Heidelberg (2010)
4. Hearst, M.A.: Automatic Acquisition of Hyponyms. Technical Report. University of California at Berkeley, Berkeley, CA, USA (1992)
5. Ruiz-Casado, M., Alfonseca, E., Castells, P.: Automatising the learning of lexical patterns: an application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. Data and Knowledge Engineering 61(3), 484–499 (2007)
6. Snow, R., Jurafsky, D., Ng, A.Y.: Learning syntactic patterns for automatic hypernym discovery. In: Proceedings of NIPS 17 (2005)
7. Alani, H.: Position paper: ontology construction from online ontologies. In: Proceedings of the 15th International Conference on World Wide Web (WWW 2006). ACM, New York (2006)
8. Ramakrishnan, C., Mendes, P.N., Wang, S., Sheth, A.P.: Unsupervised Discovery of Compound Entities for Relationship Extraction. In: Gangemi, A., Euzenat, J. (eds.) EKAW 2008. LNCS (LNAI), vol. 5268, pp. 146–155. Springer, Heidelberg (2008)
9. Zouaq, A., Gagnon, M., Ozell, B.: Semantic Analysis using Dependency-based Grammars and Upper-Level Ontologies. International Journal of Computational Linguistics and Applications 1(1-2), 85–101 (2010)
10. Kim, J., Kim, P., Chung, H.: Ontology construction using online ontologies based on selection, mapping and merging. IJWGS 7(2), 170–189 (2011)
11. Cimiano, P.: Ontology Learning and Population from Text: algorithm, evaluation and application. Springer (2006)
12. Maynard, D., Funk, A., Peters, W.: Using Lexico-Syntactic Ontology Design Patterns for ontology creation and population. In: WOP 2009 – ISWC Workshop on Ontology Patterns, Washington (2009)
13. D'Aquin, M., Kronberger, G., Suárez-Figueroa, M.: Combining Data Mining and Ontology Engineering to enrich Ontologies and Linked Data. In: Workshop: Knowledge Discovery and Data Mining Meets Linked Open Data - Know@LOD at Extended Semantic Web Conference, ESWC (2012)