Early Steps Toward Web-Scale Information Extraction with LODIE

Anna Lisa Gentile, Ziqi Zhang, Fabio Ciravegna

■ *Information extraction (IE) is the technique for transforming unstructured textual data into a structured representation that can be understood by machines. The exponential growth of the web generates an exceptional quantity of data for which automatic knowledge capture is essential. This work describes the methodology for web-scale information extraction in the linked open data information-extraction (LODIE) project and highlights results from the early experiments carried out in the initial phase of the project. LODIE aims to develop information-extraction techniques able to scale at web level and adapt to user information needs. The core idea behind LODIE is the usage of linked open data, a very large-scale information resource, as a ground-breaking solution for IE, which provides invaluable annotated data on a growing number of domains. This article has two objectives, first, describing the LODIE project as a whole and depicting its general challenges and directions; and second, describing some initial steps taken toward the general solution, focusing on a specific IE subtask, wrapper induction.*

Extracting information from a gigantic data source such as the web has been considered a major research challenge, and over the years many different approaches (Etzioni et al. 2004; Banko et al. 2007; Carlson et al. 2010; Freedman and Ramshaw 2011; Nakashole, Theobald, and Weikum 2011) have been proposed. Nevertheless, the current state of the art has mainly addressed tasks for which resources for training are available (for example, the TAP ontology in the paper by Etzioni and colleagues [2004]) or use generic patterns to extract generic facts (for example, Banko et al. [2007]; OpenCalais.com). The limited availability of resources for training has so far prevented the study of the generalized use of large-scale resources to port to specific user information needs. The linked open data information-extraction (LODIE) project focuses on the study of IE models and algorithms able to perform efficient user-centered web-scale learning by exploiting linked open data (LOD). In this article we will highlight the initial steps of the LODIE project, focusing on a specific IE task, wrapper induction (WI), which consists of automatically learning wrappers for uniform web pages, that is, pages from one website, usually generated with the same script and all describing the same type of entity. We show results on the WI task, exploiting linked data obtained from DBpedia as learning material. Linked data is a recommended best practice for exposing, sharing, and connecting data using URIs and RDF.[1] LOD is ideally suited for supporting web-scale IE adaptation because it is very large scale, constantly growing, covering multiple domains, and being used to annotate a growing number of pages that can be exploited for training. Current approaches to using LOD for web-scale IE are limited in scope to recognizing tables (Mulwad et al. 2010) and extraction of specific answers from large corpora (Balog and Serdyukov 2011), but a generalized approach to the use of LOD for training large-scale IE is still missing. The overall aim of LODIE is to study how imprecise, redundant, and large-scale resources like LOD can be used to support web-scale user-driven IE in an effective and efficient way. The idea behind the project is to adapt IE methods to detailed user information needs in a completely automated way, with the objective of creating very large domain-dependent and task-dependent knowledge bases. The underlying assumption is that LOD actually contains useful ontologies for the extraction tasks. The goal of this article is to present initial steps toward this direction, using wrapper induction as an example IE task.

Related Work

Adapting IE methods to web scale implies dealing with two major challenges: large scale and lack of training data. Traditional IE approaches apply learning algorithms that require large amounts of training data, typically created by humans. However, creating such learning resources at web scale is infeasible in practice.

Typical web-scale IE methods adopt a lightweight iterative learning approach, in which the amount of training data is reduced to a handful of manually created examples called seed data. These are searched in a large corpus to create an annotated data set, whereby extraction patterns are generalized using some learning algorithms. Next, the learned extraction patterns are reapplied to the corpus to extract new instances of the target relations or classes. Mostly these methods adopt bootstrapping techniques, where the newly learned instances are selected to seed the next round of learning, using some measures to assess their quality in order to control noisy data. Two well-known earlier systems in this area are Snowball (Agichtein et al. 2001) and KnowItAll (Etzioni et al. 2004, Banko et al. 2007). Snowball iteratively learns new instances of a given type of relation from a large document collection, while KnowItAll learns new entities of predefined classes from the web. Both have inspired a number of more recent studies, including StatSnowball (Zhu 2009), ExtremeExtraction (Freedman and Ramshaw 2011), NELL (Carlson et al. 2010), and PROSPERA (Nakashole, Theobald, and Weikum 2011). Some interesting directions undertaken by these systems include exploiting background knowledge in existing knowledge bases or ontologies to infer and validate new knowledge instances, and learning from negative seed data. While these systems learn to extract predefined types of information based on (limited) training data, the TextRunner (Banko et al. 2007) system proposes an "open information-extraction" paradigm, which exploits generic patterns to extract generic facts from the web for unlimited domains without predefined interests.

The emergence of LOD has opened an opportunity to reshape web-scale IE technologies. The underlying multibillion triple store[2] and increasing availability of LOD-based annotated web pages (for example, RDFa) can be invaluable resources to seed learning. Researchers are starting to consider the use of LOD for web-scale information extraction. However, so far research in this direction has just taken off and the use of linked data is limited. Mulwad and colleagues (2010) proposed a method to interpret tables based on linked data and extract new instances of relations and entities from tables. The TREC2011 evaluation on the related-entity finding task (Balog and Serdyukov 2011) has proposed using LOD to support answering generic queries in large corpora. While these are relevant to our research, a full, user-driven complex IE task based on LOD is still to come.

While framing this article under the general information-extraction methods, we will focus on a specific task of IE, *wrapper induction,* which we will use to showcase the initial ideas on the usage of LOD for IE. Wrapper induction (Kushmerick, Weld, and Doorenbos 1997; Muslea, Minton, and Knoblock 2003; Dalvi, Bohannon, and Sha 2009; Dalvi, Kumar, and Soliman 2011; Wong and Lam 2010) is the task of automatically learning wrappers using a collection of manually annotated web pages as training data. It generally addresses extracting data from detail web pages (Carlson and Schafer 2008), which are pages corresponding to a single data record (or entity) of a certain type or concept (also called vertical in the literature), and renders various attributes of that record in a human-readable form. An extensive range of work has been carried out to study wrapper induction in the past, and an extensive survey can be found in the paper by Doan, Halevy, and Ives (2012). However, the task remains challenging for several reasons. First, wrappers are typically induced based on training examples, which are manually labelled web pages of particular websites.

Creating such annotations requires significant human effort and remains a bottleneck in the wrapper induction process (Wong and Lam 2010; Hao et al. 2011). Second, wrappers are typically learned specific to a website and largely depend on structural consistency. Porting wrappers across websites often requires relearning (Wong and Lam 2010), and even very slight change in structures can cause wrappers to break. Although recent studies (Carlson and

Schafer 2008; Dalvi, Bohannon, and Sha 2009; Dalvi, Kumar, and Soliman 2011; Hao et al. 2011) have focused on addressing these two issues, these methods still depend on manually labelled examples to train a wrapper, while in some cases (Dalvi, Bohannon, and Sha 2009), even more training data is required to enhance wrapper robustness. To alleviate human effort, some unsupervised methods are proposed to first cluster web pages that share similar structures (for example, Blanco et al. 2011), and then deduce a shared template for each cluster of web pages. Two well-known studies in this stream are RoadRunner (Crescenzi and Mecca 2004) and EXALG (Arasu and Garcia-Molina 2003). However, such methods do not recognize the semantics of the extracted data (that is, attributes), but rely on human effort to identify attribute values from the extracted content.

The second limitation of wrappers is that they often are very specific and therefore are inflexible and not robust enough to cope with variations in the structures of web pages. It is recognized that even a very slight change in the underlying structure of web pages can cause the wrappers to break and have to be relearned. This is often referred to as the "wrapper breakage" problem (Dalvi, Bohannon, and Sha 2009; Parameswaran et al. 2011). As suggested by Gulhane and colleagues (2011), wrappers learned without robustness considerations had an average life of 2 months, with, on average, 1 out of every 50 wrappers breaking every day. Thus, research in recent years has focused on developing robust wrapper induction approaches (Dalvi, Bohannon, and Sha 2009; Dalvi, Kumar, and Soliman 2011) and methods that are general across attributes, verticals, and websites (Muslea, Minton, and Knoblock 2003; Hao et al. 2011).

While such methods have been shown to improve robustness of wrappers as well as reduce the amount of manual annotations for training, they still require seed web pages to be annotated. Hao, Cai, Pang, and Zhang (2011) for instance require at least one website to be annotated for each vertical.

LODIE — User-Centered Web-Scale IE

In LODIE we propose to develop an approach to web-scale IE that enables fully automated adaptation to specific user needs. LOD will provide ontologies to formalize the user information needs and will enable seeding learning by providing instances (triples) and web pages formally annotated through RDFa or microformats. Such background knowledge will be used to seed semisupervised web-scale learning.

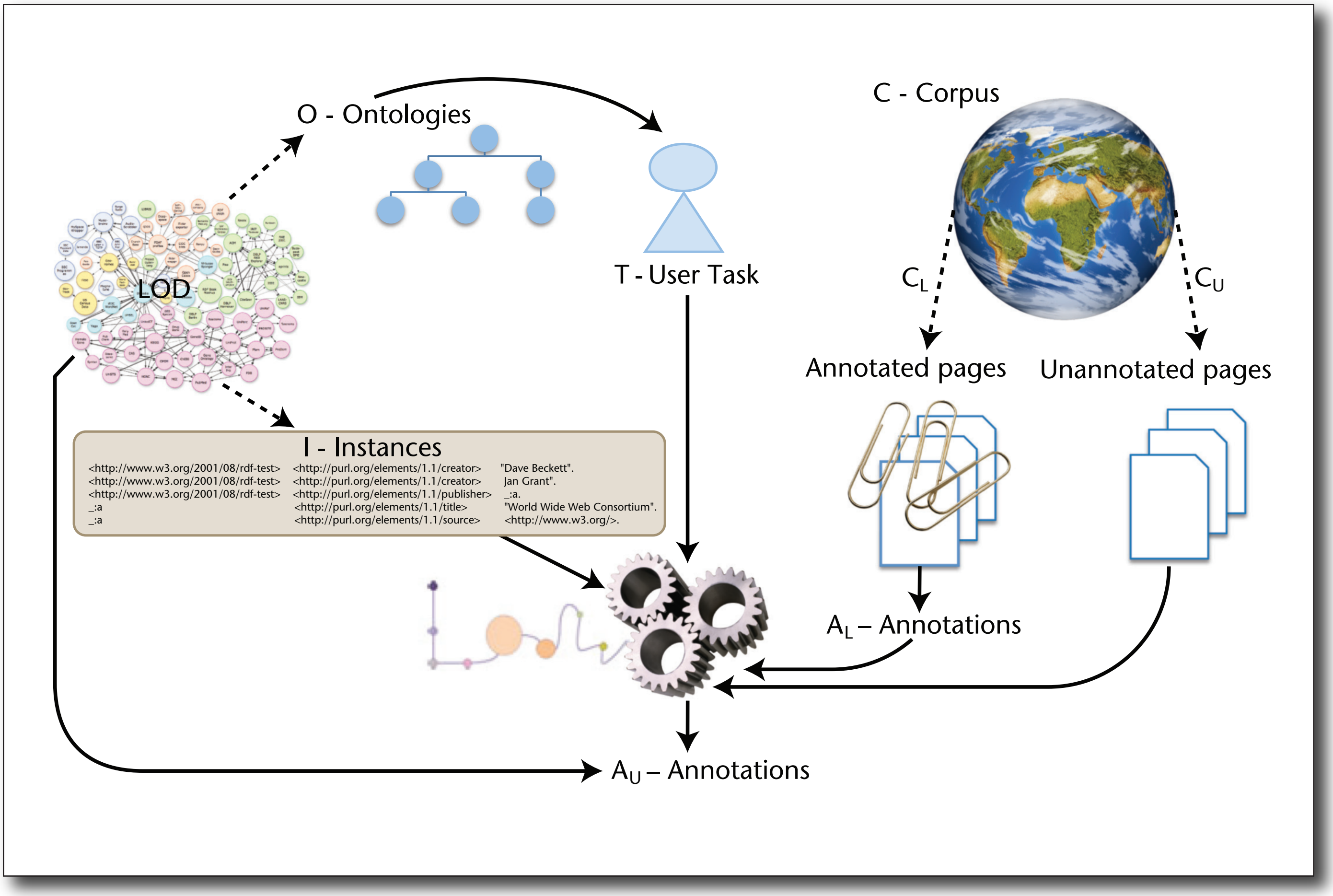
The use of an uncontrolled and constantly evolving community-provided set of independent web resources for large-scale training is totally untapped in the current state of the art. Research has shown that the relation between the quantity of training data and learning accuracy follows a nonlinear curve with diminishing returns (Thompson, Califf, and Mooney 1999). On LOD the majority of resources are created automatically by converting legacy databases with limited or no human validation; thus errors are present (Auer et al. 2009). Similarly, community-provided resources and annotations can contain errors, imprecision (Lopez et al. 2010), spam, or even deviations from standards (Halpin, Hayes, and McCusker 2010). Also, large resources can be redundant, that is, contain a large number of instances that contribute little to the learning task, while introducing considerable overhead. Very regular annotations present very limited variability, and hence high overhead for the learners (which will have to cope with thousands of examples that provide little contribution) and the high risk of overfitting the model.

The key ideas behind LODIE are (1) the formilization of user requirements for web-scale IE through LOD; (2) the usage of LOD data to seed learning; and (3) the development of multistrategy web-scale learning methods robust to noise. While we explored initial methods to formalize user needs (Zhang et al. 2013), in this article we summarise initial results on ideas (2) and (3), presenting experiments on the usage of LOD as training data for the task of wrapper induction. As the ultimate goal of LODIE, all results of IE will be published and integrated into the LOD; therefore each one will need to be assigned a URI, that is, a unique identifier. We call this step *disambiguation* (Gentile et al. 2010). This aspect has not yet been tackled in LODIE, but we will explore methods with minimum requirements in computational terms such as simple feature-overlapping-based methods (Banerjee and Pedersen 2002) and string distance metrics (Lopez et al. 2010).

LODIE — Overall Methodology

We define web-scale IE as a tuple: $< T, O, C, I, A >$ where: T is the formalization of the user information needs (that is, an IE task); O is the set of ontologies on the LOD. C is a large corpus (typically the web), which can be annotated already in part (C_L) with RDFa / microformats; we refer to the unannotated part as C_U. I represents a collection of instances (knowledge base) defined according to O; I_L is a subset of I containing instances already present on the LOD; I_U is the subset of I containing all the instances generated by the IE process when the task is executed on C. A is a set of annotations and consists of two parts: A_L are found in C_L, and A_U are created by the IE process; A_U can be the final set or the intermediate sets created to reseed learning. Figure 1 sketches the general workflow of LODIE.

As an example throughout the article we will consider the following: A user is interested in films, with their titles and directors. She starts by exploring ontologies on LOD and selects concepts and attrib-



```
<http://www.w3.org/2001/08/rdf-test>   <http://purl.org/elements/1.1/creator>     "Dave Beckett".
<http://www.w3.org/2001/08/rdf-test>   <http://purl.org/elements/1.1/creator>      Jan Grant".
<http://www.w3.org/2001/08/rdf-test>   <http://purl.org/elements/1.1/publisher>   _:a.
_:a                                     <http://purl.org/elements/1.1/title>        "World Wide Web Consortium".
_:a                                     <http://purl.org/elements/1.1/source>       <http://www.w3.org/>.
```

Figure 1. High-Level LODIE Workflow.

utes that represent her needs. These will define the task *T*. I_L in this case consists of all the triples identifying film titles and directors, retrievable on the LOD. C_L is the portion of web pages containing annotations (for example, RDFa / microformats) about film titles and directors. Given C_U, web pages without annotations, LODIE will try and identify instances of film titles and directors. The generated annotations (A_U) can either refer to instances already present in the LOD (I_L) or to novel facts (I_U). In the second case the new triples (I_U) will be published to the LOD.

User Needs Formilization

The first requirement for adapting web-scale IE to specific user needs is to support users in formalizing their information needs in a machine-understandable format. Formally we define the user needs as a function: $T = f(O) \rightarrow OL$ identifying a view on the LOD ontologies[3] describing the information-extraction task. *T* will be materialized in the form of an OWL ontology. In the running example the user wants to collect a number of films, with their titles and directors. The relevant ontology OL in this case will include the concept[4] and the properties title[5] and director.[6]

The naive way of defining *T* is to manually identify relevant ontologies and concepts on the LOD and define a view on them. The solution we propose is to exploit statistical knowledge patterns (SKP) (Zhang et al. 2013) as a gate to the LOD ontologies. An SKP is an ontological view over a class (defined in a reference ontology in the LOD), and captures and summarises the usage of that class in data. An SKP is represented and stored as an OWL ontology. Each SKP contains properties and axioms involving the main class derived from a reference ontology and properties and axioms involving the main class that are not expressed in the reference ontology but that can be induced from statistical measures on statements published as linked data. Although so far we only addressed synonymity of relations (Zhang et al. 2013), synonymity of concepts in different data sets can be included (Parundekar, Knoblock, and Ambite 2012).

At this stage we do not propose a systematic methodology to define *T*, and we still require the user to manually define *T*, but we aim to ease the task by providing SKPs, which we consider as building blocks to add additional information to the underneath ontologies.

Gathering Training Seeds

A set of triples I_L relevant to the users' needs are identified as side effect of the definition of T: they can be retrieved from existing LOD knowledge bases associated with the types in T, using available SPARQL endpoints. Following the running example, DBpedia SPQRQL endpoint[7] can be queried for all titles (dbpedia.org/property/name) and directors (dbpedia.org/ontology/director) of resources of type schema.org/Movie. Annotations, A_L, can be also retrieved exploiting RDFa and microformat in web pages, relevant to the types in T (if available). In our film example, semantic annotations can be extracted from pages of the IMDB web site,[8] among others. Further candidates, A_U, can be identified by searching the web for linguistic realization of the triples I_L, which means text in web pages matching values within the triples. In our example, let's consider the triple in I_L with subject dbpedia.org/resource/The Godfather, predicate dbpedia.org/property/name and object "The Godfather"@en. The text in the object "The Godfather" can be used to retrieve additional possible candidates in web pages. The annotations are used by the multistrategy learning process to create new candidate annotations and instances. Some learning tasks won't be robust to noise in the training data, and therefore filtering will be needed; other tasks will be able to handle noise (Gentile et al. 2013). In LODIE we did not yet carry systematic experiments on data filtering, but our intuition is that to obtain good quality seeds we need to obtain a good trade-off between consistency and variability of examples. We will cast filtering as a problem of detecting noise in training data (Jiang and Zhou 2004, Valizadegan and Tan 2007).

Multistrategy Learning

The seed data identified and filtered in the previous steps are submitted to a multistrategy learning method, which is able to work in different ways according to the type of web pages the information is located in: (1) a model M_S able to extract from regular structures such as tables and lists; (2) a model M_W wrapping very regular websites generated by backing databases; and (3) a model M_T for information in natural language based on lexical-syntactic extraction patterns. The M_S and M_T models have not been yet implemented in LODIE, while we did implement a prototype for M_W (Gentile et al. 2013). As for M_S we are currently working on an unsupervised approach able to interpret types, entities, and relations expressed in tables using both linked data knowledge bases and RDFa / microdata annotations within web pages. The method follows a two-step bootstrapping manner, where the first phase (learn) interprets tables based on a limited sample from tables to ensure efficiency and the second phase (update) revises and extends the interpretation to ensure accuracy. As for M_T we will base the strategy on learning shallow patterns. As opposed to approaches based on complex machine-learning algorithms, for example, random walks (Iria, Xia, and Zhang 2007), we will focus on lexical-syntactic shallow pattern-generalization algorithms. The patterns will be generalized from the textual context of each $a \in A$ and will be based on features such as words (lexical), part of speech (syntactic), and expected semantics such as related entity classes. The patterns are then applied to other web pages to create new candidate annotations. The M_W model is implemented as an unsupervised wrapper induction system. The basic idea is to start with seed gazetteers from LOD and learn wrappers based on the occurrence of gazetteer elements on web pages in a brute-force manner. We describe the design of the M_W model in detail in the following section.

At the end of this process, we concatenate the candidate annotations extracted by each learning strategy, M_S, M_W, and M_T, and create a collection of candidates $a \in A_U$. These will refer to instances already known (I_L) as well as new instances (I_U).

Wrapper Induction

Wrapper induction (WI) is the task of automatically learning wrappers for uniform web pages, that is, all generated with the same script and all describing the same type of entity. Given as input: a concept c_i, some of its attributes $\{a_{i,1}, \ldots, a_{i,k}\}$, a set of uniform web pages W_{ci} describing entities of the concept c_i; the goal of WI is to retrieve values for attributes $a_{i,k}$ for each entity of type c_i, in W_{ci}. The set of concepts $C = \{c_1, \ldots, c_i\}$ and the set of attributes $\{a_{i,1}, \ldots, a_{i,k}\}$ come from the the definition of T, which is the output of the user needs formilization phase.

We propose a three-step approach, where we (1) build pertinent dictionaries to (2) annotate web pages and (3) discover the structural patterns that encapsulate the target information. The dictionary-building phase corresponds to the gathering-training-seeds phase in the general architecture. Obtaining dictionaries from the web is a topic that has been tackled in previous research, for example, Michelson and Knoblock (2008) and Zhang and Iria (2009). In this work we do not implement any novel technique for it, but we simply exploit linked data. Each dictionary is identified as a side effect of the definition of T. In practice data is obtained querying available SPARQL endpoints, with a pertinent query for each concept c_i – attribute $a_{i,k}$ pair, thus obtaining a dictionary $d_{i,k}$ for each attribute $a_{i,k}$ of each concept c_i. In the running example, one of the queries could be "SELECT DISTINCT ?title WHERE{ ?film a <schema.org/Movie>;<dbpedia.org/property/name> ?title . FILTER (langMatches(lang(?title), 'EN')).}."

The dictionary-generation process is completely independent from the data in W_{ci}. No a priori knowledge about the data is introduced to this process and thus the dictionary $d_{i,k}$ is unbiased and universal for any extraction tasks concerning the pair $c_i - a_{i,k}$.

Concept	WS	WP	Attributes
Book	10	20000	title (t), author (a), ISBN-13 (i), publisher (p), publish-date (pd)
Movie	10	20000	title (t), director (d), genre (g), rating (r)
NBA player	10	4405	name (n), team (t), height (h), weight (w)
Restaurant	10	20000	name (n), address (a), phone (p), cuisine (c)
University	10	16705	name (n), phone (p), website (w), type (t)

Table 1. Statistics of the Selected Gold Standard Data Set (Hao et al. 2011).

WS is the number of websites and WP is the total number of web pages for the concept.

Each web page in W_{ci} is parsed into a DOM tree, and for each leaf node containing text we save its text value and its *xpath*.[9] Then to generate the annotations for the attribute $a_{i,k}$ of c_i, the text content of each node is matched against the dictionary $d_{i,k}$. If there is an exact match between the text content of a node and any item in the dictionary, the annotation is saved as a pair <*xpath*, *text value*>. We assume that the values to extract are fully and exactly contained in a page node. Although this is a strong assumption, which will be relaxed in future versions, it is a common feature among websites for observed attributes.

The annotation step produces a set of <*xpath*, *text value*> pairs for each website. There are two problems with these annotations. First, due to the incompleteness of the autogenerated dictionaries, the annotation process may not cover the entire data set and the number of false negatives can be large (that is, low recall). However, we expect the large dictionaries to cover more than enough to learn useful wrappers for the attributes. Second, entries in the dictionaries can be ambiguous (for example, *Home* is a book title that matches part of navigation paths on many web pages) but annotation does not involve disambiguation. The intuition is that useful *xpath*s will be likely to match a bigger variety of dictionary entries, on a sufficiently large sample of web pages. For a particular attribute $a_{i,n}$ and a website collection W_{ci}, starting from all generated annotations <*xpath*, *text value*> we create a map containing as keys all distinct *xpath*s and as values all distinct *text value*s corresponding to each *xpath*. Based on the hypothesis of structural consistency in a website, we expect the majority of true positives to share the same or similar *xpath*. Also, since an attribute is likely to have various distinct values, we rank the map with decreasing number of values for each *xpath* and consider the top-ranked items to be useful *xpath*s for extracting the attribute $a_{i,n}$ on this website collection. In this experiment we always induce a single *xpath* per attribute, therefore we always select the highest ranked *xpath* to be the wrapper for the attribute on the specific website.

The web page annotation and *xpath* identification are repeated for every attribute on every website collection, creating a wrapper for each attribute-website pair. Finally, each wrapper is applied to reannotate the website for the corresponding attributes.

Experiments

The WI strategy comprehends three steps: (1) gathering relevant dictionaries, (2) automatically generating annotations based on those dictionaries, (3) inducing the wrappers using the annotations. The hypothesis is that the method is successful in the presence of good quality dictionaries. To test this hypothesis we performed an experiment where good quality dictionaries are artificially generated, specifically tailored to the data in each W_{ci}. In this way we tested if the brute force pattern induction methodology was likely to succeed, independently from the usage of LOD as the background source. We performed steps (2) and (3) using these ideal dictionaries. This experiment is reported in Gentile et al. (2013), and the induced patterns produced good extraction results, with overall F-measure of 80 percent on a publicly available data set (Hao et al. 2011).

The data set consists of around 124,000 pages collected from 80 websites. These websites are related to eight concepts, including *Autos, Books, Cameras, Jobs, Movies, NBA Players, Restaurants,* and *Universities,* 10 websites per concept, with 200 to 2000 detail pages per website. Each concept has three to five common attributes selected for the extraction task. Table 1 shows the statistics of the data set, where WS shows the number of different websites and WP shows the total number of web pages. A groundtruth for the WI task is also provided by Hao et al. (2011). It consists of one file for each attribute-website pair, listing all possible attribute values found on the website is generated. The values have been obtained by using a few handcrafted regular expressions over each website.

The goal of the experiment presented in this section is to test if LOD can serve the generation of suitable dictionaries for our proposed methodology.

Starting with the set and concepts and attributes of our reference data set (see table 1) we translated each of them in a task T. In the general LODIE workflow, the user task definition is performed by the user, who selects the concepts directly from available ontologies in the LOD; therefore all concepts c_i and attributes $a_{i,k}$ are already represented as URI from the LOD. For the sake of the experiment, we manually performed this mapping for concepts and attributes in our reference data set (in table 1), manually searching the LOD for suitable ones. We were able to map five of the original eight concepts to LOD ontologies. For each mapped concept-attribute pair, we queried DBpedia SPARQL endpoint to retrieve all unique objects of triples with subject of type c_i and with $a_{i,k}$ as property. Table 2 reports statistics of generated dictionaries for mapped concepts.

The data extracted by our wrappers is compared against the groundtruth values provided by Hao et al. (2011). To calculate the performance metrics we followed the evaluation methodology proposed by Hao et al. (2011). Considering that their method is designed to extract only one value per attribute, they suggest that a prediction be counted correct *(page hit)* as long as at least one of the answers in the ground truth is extracted. Precision is then the number of page hits divided by the number of pages for which the method extracts values. Recall is the number of page hits divided by the number of pages in the groundtruth containing attributes values. We do report F1 measure, which takes into account both P and R. Figure 2 shows the average F1 of the extraction using the generated wrappers for each concept-attribute across all websites. Take *book*, for example; figure 2 shows that the wrappers for the *title* (*t*) attribute induced by our method can extract true positives from all of the 10 websites, with an average F-measure of 90 percent.

Table 3 compares the average of results of our method (LOD dictionaries) against the same method applied starting from Ideal Dictionaries (figures obtained from Gentile et al. [2013]) and against the method by Hao et al. (2011), from which we also obtained the data set. Hao and colleagues designed the WI method based on two types of features, which they call weak and strong features. Weak features are general across attributes, verticals, and websites, and they are used to identify a large amount of candidate attribute values. Strong features, which are site specific, derived in an unsupervised manner, are exploited to boost the true values. They require users to manually annotate one website for each concept, from which they can perform the initial collection of weak features. Our method does not outperform Hao's (Hao et al. 2011), but has the advantage of being totally unsupervised. The comparison between

Concept	Attribute	LD
University	phone	283
	website	12,930
	name	13,144
	type	
Book	isbn 13	39,112
	author	13,060
	title	37,485
	publication date	3048
	publisher	520
Movie	genre	114
	title	57,292
	mpaa rating	2
	director	16,079
Restaurant	phone	
	cuisine	72
	address	37
	name	312
NBA player	weight	
	height	
	name	9194
	team	677

Table 2. Attribute Dictionaries Statistics.

LD is the number of items in the dictionary.

Concept	Hao et al.	Ideal Dictionaries	LOD Dictionaries
auto	0.71	0.94	
book	0.87	0.85	0.78
camera	0.91	0.76	
job	0.85	0.82	
movie	0.79	0.86	0.76
nbaplayer	0.82	0.9	0.87
restaurant	0.96	0.89	0.69
university	0.83	0.96	0.91

Table 3. Comparison of F-Measure per Concept.

Hao et al. (2011) refer to figures reported in the paper From One Tree to a Forest; *Ideal Dictionaries* refers to our method applied using ad hoc dictionaries, reported as topline experiment (Gentile et al. 2013).

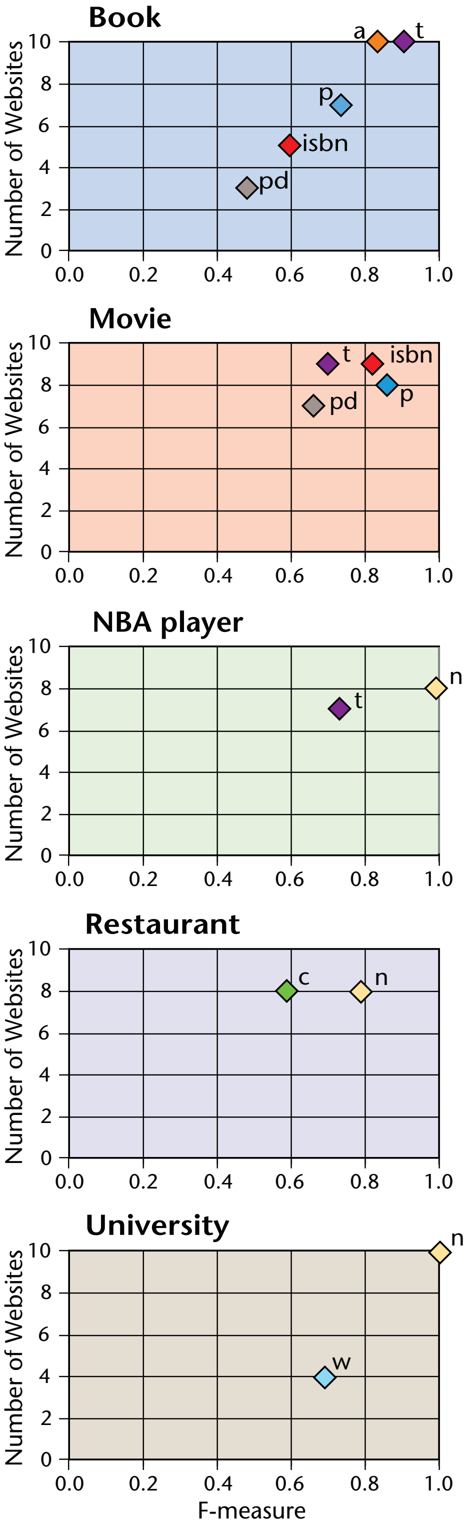


Figure 2. Results of LD-Based WI for Each Attribute of a Concept.

The *x*-axis Reports the F Measure, while the *y*-axis indicates the number N of covered websites in the test set. The labels on each point are shorthand to indicate the attributes (keys in table 1).

LOD dictionaries and ideal dictionaries hints that the quality of dictionaries has an impact on the wrapper induction process, as the figures of the method using LD-based dictionaries are always below the same method using ideal dictionaries. An error analysis phase revealed that some failure cases are related to the fact that some dictionaries are either very small or do not contain any semantically correct seed at all. Indeed, even with a high presence of noise, the method is robust as long as the dictionary contains a few good seeds. As an example, the dictionary that we generate for the attribute genre of the concept movie is rather semantically dubious. If querying for the property dbpedia.org/property/ genre of the concept schema.org/Movie, returned values mostly refer to the genre of the movie soundtrack than the genre of the movie, which is a result of the triple extraction process from Wikipedia to DBpedia (Kobilarov et al. 2009). We manually checked the dictionary entries (114) for genre. A total of 111 refer to music genre (for example, indie pop, country music, drum and bass, groove metal, indie rock,and others), while only 3 of them refer to actual movie genre (philosophical fiction, thriller, western). Nevertheless, the performance of the method for this attribute is perfectly comparable with one obtained with the ideal dictionary, thus showing that the intuition of using LOD as training material for IE task goes in a promising direction.

One limitation of our experiment is the usage of a single resource from LOD (DBpedia) rather than exploiting the LOD as a whole. As work in progress and in line with the general directions of LODIE, we are currently extending all preliminary methods to the usage of multiple resources.

Conclusion

LODIE is a project that addresses complex challenges that we believe are novel and of high interest to the scientific community. It is timely because for the first time in the history of IE a very large-scale information resource is available, covering a growing number of domains and of the very recent interest in the use of linked data for web extraction. While the LODIE project as a whole tackles diverse challenges, in this article we focused on how to obtain training data from the LOD and use them for an IE task. We experimented with the use of linked data dictionaries to automatically generate annotations for learning. The generated annotations, despite the presence of noise, are valuable for certain tasks. We carried out an experiment for the wrapper induction task, and the automatically generated annotations led to comparable results with the state of the art. Although this is an encouraging result, we expect the noise to be more problematic for learning tasks that are more complicated than WI. In future, while the development of the remaining components described in the LODIE framework will be carried forward, special focus will be placed on generic methods able to port to different linked data sets and efficient methods able to cope with the large-scale data, possibly by combining data sampling and filtering techniques.

Acknowledgments

Part of this research has been sponsored by the EPSRC funded project LODIE, EP/J019488/1.

Notes

[1] www.linkeddata.org.

[2] www4.wiwiss.fu-berlin.de/lodcloud.

[3] A view is a smaller ontology only including concepts and relations that can describe the user needs.

[4] schema.org/Movie.

[5] dbpedia.org/property/name.

[6] dbpedia.org/ontology/director.

[7] for example, dbpedia.org/sparql.

[8] www.imdb.com.

[9] www.w3.org/TR/xpath.

References

Agichtein, E.; Gravano, L.; Pavel, J.; Sokolova, V.; and Voskoboynik, A. 2001. Snowball: A Prototype System for Extracting Relations from Large Text Collections. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data,* 612. New York: Association for Computing Machinery

Arasu, A., and Garcia-Molina, H. 2003. Extracting Structured Data from Web Pages. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data,* 337–348. New York: Association for Computing Machinery

Auer, S.; Dietzold, S.; Lehmann, J.; Hellmann, S.; and Aumueller, D. 2009. Triplify: Light-weight Linked Data Publication from Relational Databases. In *Proceedings of the 18th International Conference on World Wide Web,* WWW'09, 621–630. New York: Association for Computing Machinery.

Balog, K., and Serdyukov, P. 2011. Overview of the TREC 2010 Entity Track. In *Proceedings of the Nineteenth Text REtrieval Conference, TREC 2010*. Gaithersburg, MD: National Institute of Standards and Technology.

Banerjee, S., and Pedersen, T. 2002. An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In *Proceedings of the Third International Conference on Computational Linguistics and Intelligent Text Processing,* CICLing'02, 136–145. Berlin: Springer.

Banko, M.; Cafarella, M.; Soderland, S.; Broadhead, M.; and Etzioni, O. 2007. Open Information Extraction for the Web. In *Proceedings of the 20th International Joint Conference on Artifical intelligence,* IJCAI'07, 2670–2676. Palo Alto, CA: AAAI Press.

Blanco, R.; Halpin, H.; Herzig, D.; and Mika, P. 2011. Entity Search Evaluation over Structured WeData. In *Proceedings of the 1st International Workshop on Entity-Oriented Search at SIGIR 2011.* Delft, Netherlands: Delft University of Technology.

Carlson, A., and Schafer, C. 2008. Bootstrapping Information Extraction from Semi-Structured Web Pages. In *Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases* – Part I, ECML-PKDD'08, 195–210. Berlin: Springer-Verlag.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Jr, E. R. H.; and Mitchell, T. M. 2010. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence* (AAAI), 1306–1313. Palo Alto, CA: AAAI Press.

Crescenzi, V., and Mecca, G. 2004. Automatic Information Extraction from Large Websites. *Journal of the ACM* 51(5): 731–779. dx.doi.org/10.1145/1017460.1017462

Dalvi, N.; Bohannon, P.; and Sha, F. 2009. Robust Web Extraction: An Approach Based on a Probabilistic Tree-Edit Model. In *Proceedings of the 35th SIGMOD International Conference on Management of Data*. New York: Association for Computing Machinery.

Dalvi, N.; Kumar, R.; and Soliman, M. 2011. Automatic Wrappers for Large Scale Web Extraction. *Proceedings of the VLDB Endowment* 4(4): 219–230. dx.doi.org/10.14778/1938545.1938547

Doan, A.; Halevy, A. Y.; and Ives, Z. G. 2012. *Principles of Data Integration*. San Francisco: Morgan Kaufmann Publishers.

Etzioni, O.; Cafarella, M.; Downey, D.; Kok, S.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2004. Web-Scale Information Extraction in KnowItAll (Preliminary Results). In *Proceedings of the 13th International Conference on World Wide Web,* WWW'04, 100–110. New York: Association for Computing Machinery.

Freedman, M., and Ramshaw, L. 2011. Extreme Extraction: Machine Reading in a Week. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing,* EMNLP 2011, 1437–1446. Stroudsberg, PA: Association for Computational Linguistics.

Gentile, A. L.; Zhang, Z.; Augenstein, I.; and Ciravegna, F. 2013. Unsupervised Wrapper Induction Using Linked Data. In *Proceedings of the Seventh International Conference on Knowledge Capture*, K-CAP'13, 41–48. New York: Association for Computing Machinery.

Gentile, A. L.; Zhang, Z.; Xia, L.; and Iria, J. 2010. Semantic Relatedness Approach for Named Entity Disambiguation. In *Digital Libraries*, volume 91 in *Communications in Computer and Information Science*, 137–148. Berlin: Springer.

Gulhane, P.; Madaan, A.; Mehta, R.; Ramamirtham, J.; Rastogi, R.; Satpal, S.; Sengamedu, S. H.; Tengli, A.; and Tiwari, C. 2011. Web-Scale Information Extraction with Vertex. In *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering,* ICDE'11, 1209–1220. Los Alamitos, CA: IEEE Computer Society.

Halpin, H.; Hayes, P.; and McCusker, J. 2010. When Owl: Sameas Isn't the Same: An Analysis of Identity in Linked Data. In *Proceedings of 9th International Semantic Web Conference*, Lecture Notes in Computer Science Volume 6496 ISWC'10, 305–320. Berlin: Springer.

Hao, Q.; Cai, R.; Pang, Y.; and Zhang, L. 2011. From One Tree to a Forest: A Unified Solution for Structured Web Data Extraction. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval,* SIGIR'11, 775–784. New York: Association for Computing Machinery.

Iria, J.; Xia, L.; and Zhang, Z. 2007. WIT: Web People Search

Disambiguation Using Random Walks. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, SemEval'07, 480–483. Stroudsburg, PA: Association of Computational Linguistics.

Jiang, Y., and Zhou, Z.-H. 2004. Editing Training Data for kNN Classifiers with Neural Network Ensemble. In *Proceedings of the International Symposium on Neural Networks,* ISNN 2004, volume 3173 of Lecture Notes in Computer Science, 356–361. Berlin: Springer.

Kobilarov, G.; Bizer, C.; Auer, S.; and Lehmann, J. 2009. DBpedia — A Linked Data Hub and Data Source for Web and Enterprise Applications. In *Proceedings of the 18th International World Wide Web Conference,* WWW'09, 1–3. New York: Association for Computing Machinery.

Kushmerick, N.; Weld, D. S.; and Doorenbos, R. B. 1997. Wrapper Induction for Information Extraction. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence,* IJCAI'97, 729–737. San Francisco: Morgan Kaufmann Publishers.

Lopez, V.; Nikolov, A.; Sabou, M.; Uren, V.; Motta, E.; and D'Aquin, M. 2010. Scaling Up Question-Answering to Linked Data. In *Proceedings of the 17th International Conference on Knowledge Engineering and Management by the Masses,* EKAW'10, 193–210. Berlin: Springer.

Michelson, M., and Knoblock, C. A. 2008. Creating Relational Data from Unstructured and Ungrammatical Data Sources. *Journal of Artificial Intelligence Research* 31(1): 543–590.

Mulwad, V.; Finin, T.; Syed, Z.; and Joshi, A. 2010. Using Linked Data to Interpret Tables. In *Proceedings of the First International Workshop on Consuming Linked Data,* COLD2010. Volume 665, CEUR Workshop Proceedings. Aachen, Germany: RWTH Aachen University.

Muslea, I.; Minton, S.; and Knoblock, C. 2003. Active Learning with Strong and Weak Views: A Case Study on Wrapper Induction. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence,* IJCAI'03, 415–420. Palo Alto: AAAI Press.

Nakashole, N.; Theobald, M.; and Weikum, G. 2011. Scalable Knowledge Harvesting with High Precision and High Recall. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining,* WSDM'11, 227–236. New York: Association for Computing Machinery

Parameswaran, A.; Dalvi, N.; Garcia-Molina, H.; and Rastogi, R. 2011. Optimal Schemes for Robust Web Extraction. In *Proceedings of the VLDB Endowment* 4(4): 219–230.

Parundekar, R.; Knoblock, C. A.; and Ambite, J. L. 2012. Discovering Concept Coverings in Ontologies of Linked Data Sources. In *Proceedings of the 11th International Conference on The Semantic Web* – Volume Part I, ISWC'12, 427–443. Berlin: Springer-Verlag.

Thompson, C. A.; Califf, M. E.; and Mooney, R. J. 1999. Active Learning for Natural Language Parsing and Information Extraction. In *Proceedings of the Sixteenth International Conference on Machine Learning,* 406–414. San Francisco: Morgan Kaufmann Publishers Inc.

Valizadegan, H., and Tan, P. 2007. Kernel Based Detection of Mislabeled Training Examples. In *Proceedings of the Seventh SIAM International Conference on Data Mining,* 309–319. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Wong, T., and Lam, W. 2010. Learning to Adapt Web Information Extraction Knowledge and Discovering New Attributes via a Bayesian Approach. *IEEE Knowledge and Data Engineering* 22(4): 523–536. dx.doi.org/10.1109/TKDE.2009.111

Zhang, Z., and Iria, J. 2009. A Novel Approach to Automatic Gazetteer Generation Using Wikipedia. In *Proceedings of the 2009 Workshop on the People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, People's Web'09, 1–9. Stroudsburg, PA: Association for Computational Linguistics.

Zhang, Z.; Gentile, A. L.; Blomqvist, E.; Augenstein, I.; and Ciravegna, F. 2013. Statistical knowledge patterns: Identifying Synonymous Relations In Large Linked Datasets. In *Proceedings of the 12th International Semantic Web Conference*. ISWC'13, Lecture Notes in Computer science volume 8218, 703–719. Berlin: Springer.

Zhu, J. 2009. StatSnowball: A Statistical Approach to Extracting Entity. In *Proceedings of the 18th International Conference on World Wide Web*, WWW'09, 101–110. New York: Association for Computing Machinery

Anna Lisa Gentile is a postdoctoral researcher in the OAK research lab of the Department of Computer Science, University of Sheffield, UK. She received a Ph.D. in computer science from the University of Bari, Italy in 2010. Her general research interests include information extraction, named entities, semantic web, and linked data. Her current research is focused on exploiting publicly available web resources (mostly linked data) to perform web-scale information extraction.

Ziqi Zhang is a postdoctoral researcher in the OAK research lab of the Department of Computer Science, University of Sheffield, UK. He received a Ph.D. in computer science from the University of Sheffield, UK in 2013. His general research interests include information extraction, disambiguation, lexical semantics, and semantic web. His current research focuses on mining and exploiting background knowledge from (mostly web-based) data resources (for example, Wikipedia, Wiktionary, DBpedia, and linked data in general) to support various NLP tasks, such as semantic relatedness, disambiguation, and information extraction.

Fabio Ciravegna is a professor of language and knowledge technologies at the University of Sheffield. His research field concerns knowledge and information management over large scale, covering three main areas: how to capture information over large scale from multiple sources and devices (the web, the social web, distributed organizational archives, mobile devices, and so on); how to use the captured information (for example, for knowledge management, business intelligence, customer analysis, management of large scale events through social media, and so on); and how to communicate the information (to final users, problem owners, and others). He is the director of the European Project WeSenseIt on citizen observatories of water and principal investigator in the EPSRC project LODIE. He is cofounder and director of K-Now Ltd, a company providing knowledge management solutions, and cofounder and scientific advisor to the Floow, an international telematics company. He holds a Ph.D. from the University of East Anglia and a doctorate from the University of Torino, Italy.