User driven Information Extraction with LODIE

Anna Lisa Gentile and Suvodeep Mazumdar

Department of Computer Science, University of Sheffield, UK
{a.gentile, s.mazumdar}@sheffield.ac.uk

Abstract. Information Extraction (IE) is the technique for transforming unstructured or semi-structured data into structured representation that can be understood by machines. In this paper we use a user-driven Information Extraction technique to wrap entity-centric Web pages. The user can select concepts and properties of interest from available Linked Data. Given a number of websites containing pages about the concepts of interest, the method will exploit (i) recurrent structures in the Web pages and (ii) available knowledge in Linked data to extract the information of interest from the Web pages.

1 Introduction

Information Extraction transforms unstructured or semi-structured text into structured data that can be understood by machines. It is a crucial technique towards realizing the vision of the Semantic Web. Wrapper Induction (WI) is the task of automatically learning wrappers (or extraction patterns) for a set of homogeneous Web pages, i.e. pages from the same website, generated using consistent templates[1]. WI methods [1,2] learn a set of rules enabling the systematic extraction of specific data records from the homogeneous Web pages. In this paper we adopt a user driven paradigm for IE and we perform on demand extraction on entity-centric webpages. We adopt our WI method [2,3] developed within the LODIE (Linked Open Data for Information Extraction) framework [4]. The main advantage of our method is that does not require manually annotated pages. The training examples for the WI method are automatically generated exploiting Linked Data.

2 State of the Art

Using WI to extract information from structured Web pages has been studied extensively. Early studies focused on the DOM-tree representation of Web pages and learn a template that wrap data records in HTML tags, such as [1,5,6]. Supervised methods require manual annotation on example pages to learn wrappers for similar pages [1,7,8]. The number of required annotations can be drastically reduced by annotating pages from a specific website and then adapting the learnt

[1] For example, a yellow page website will use the same template to display information (e.g., name, address, cuisine) of different restaurants.

rules to previously unseen websites of the same domain [9,10]. Completely unsupervised methods (e.g. RoadRunner [11] and EXALG [12]) do not require any training data, nor an initial extraction template (indicating which concepts and attributes to extract), and they only assume the homogeneity of the considered pages. The drawback of unsupervised methods is that the semantic of produced results is left as a post-process to the user. Hybrid methods [2] intend to find a tradeoff with these two limitations by proposing a supervised strategy, where the training data is automatically generated exploiting Linked Data. In this work we perform IE using the method proposed in [2,3] and follow the general IE paradigm from [4].

3 User-driven Information Extraction

In LODIE we adopt a user driven paradigm for IE. As first step, the user must define her/his information need. This is done via a visual exploration of linked data (Figure 1).

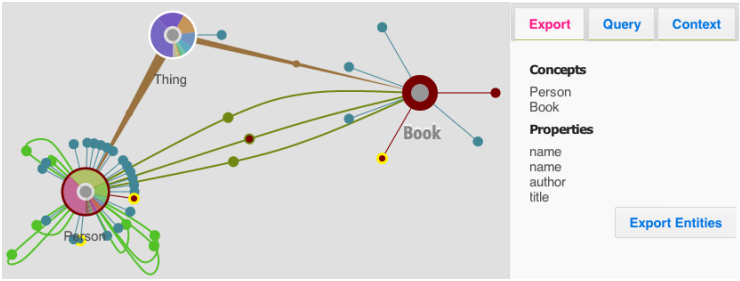

Fig. 1: Exploring linked data to define user need, by selecting concepts and attributes to extract. Here the user selected the concept *Book* and the attributes *title* and *author*. As *author* is a datatype attribute, of type *Person*, the attribute *name* is chosen.

The user can explore underlying linked data using the Affective Graphs visualization tool [13] and select concepts and properties she/he is interested in (a screenshot is shown in Figure 1). These concepts and properties get added to the side panel. Once the selection is finished, she/he can start the IE process. The IE starts with a dictionary generation phase. A dictionary $d_{i,k}$ consists of values for the attribute $a_{i,k}$ of instances of concept c_i. Noisy entries in the dictionaries are removed using a cleaning procedure detailed in [3]. As a running example we will assume the user wants to extract *title* and *author* for the concept *Book*. We retrieve from the Web k websites containing entity-pages of the concept types selected by the user, and save the pages $W_{c_i,k}$. Following the *Book* example, Barnes&Noble[2] or AbeBooks[3] websites can be used, and pages collected in $W_{book,barnesandnoble}$ and $W_{book,abebooks}$.

For each $W_{c_i,k}$ we generate a set of extraction patterns for every attribute. In our example we will produce 4 sets of patterns, one per each website and

[2] `http://www.barnesandnoble.com/`
[3] `http://www.abebooks.co.uk`

attribute. To produce the patterns we (i) use our dictionaries to generate brute-force annotations on the pages in $W_{c_i,k}$ and then (ii) use statistical (occurrence frequency) and structural (position of the annotations in the webpage) clues to choose the final extraction patterns.

Briefly, a page is transformed to a simplified page representation P_{c_i}: a collection of pairs $\langle xpath^4, \textit{text value} \rangle$. Candidates are generated matching the dictionaries $d_{i,k}$ against possible *text values* in P_{c_i} (Figure 2).

```
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[1]/H2[1]/text()[1] breaking dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/DIV[1]/H2[1]/EM[1]/text()[1] breaking dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/TABLE[10]/TBODY[1]/TR[1]/TD[3]/B[1]/A[1]/text()[1] break-
ing dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/TABLE[1]/TBODY[1]/TR[1]/TD[3]/B[1]/A[1]/text()[1]  break-
ing dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/TABLE[2]/TBODY[1]/TR[1]/TD[3]/B[1]/A[1]/text()[1]  break-
ing dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/TABLE[3]/TBODY[1]/TR[1]/TD[3]/B[1]/A[1]/text()[1]  break-
ing dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/TABLE[6]/TBODY[1]/TR[1]/TD[3]/B[1]/A[1]/text()[1]  break-
ing dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/TABLE[8]/TBODY[1]/TR[1]/TD[3]/B[1]/A[1]/text()[1]  break-
ing dawn
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[3]/DIV[3]/UL[1]/LI[2]/A[1]/text()[1] the host
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[3]/DIV[3]/UL[1]/LI[5]/A[1]/text()[1] new moon
```

Fig. 2: Example of candidates for book title for a Web page on the book "Breaking Dawn", from the website AbeBooks.

Final patterns are chosen amongst the candidates exploiting frequency information and other heuristics. Details of the method can be found in [2,3]. In the running example, higher scoring patterns for extracting book title from AbeBooks website are shown in Figure 3.

```
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[1]/H2[1]/text()[1] 329.0
/HTML[1]/BODY[1]/DIV[2]/DIV[2]/DIV[2]/DIV[4]/DIV[1]/H2[1]/EM[1]/text()[1] 329.0
```

Fig. 3: Extraction patterns for book titles from AbeBooks website.

All extraction patterns are then used to extract target values from all $W_{c_i,k}$. Results are produced as linked data, using the concept and properties initially selected by the user for representation, and made accessible to the user via an exploration interface (Figure 4), implemented using Simile Widgets[5].

A video showing the proposed system used with the running *Book* example can be found at `http://staffwww.dcs.shef.ac.uk/people/A.L.Gentile/demo/iswc2014.html`.

4 Conclusions and future work

In this paper we describe the LODIE approach to perform IE on user defined extraction tasks. The user is prompted a visual tool to explore available linked data and choose concepts for which she/he wants to mine additional material from the Web. We learn extraction patterns to wrap relevant websites and return structured results to the user.

[4] http://www.w3.org/TR/xpath/

[5] `http://www.simile-widgets.org/`

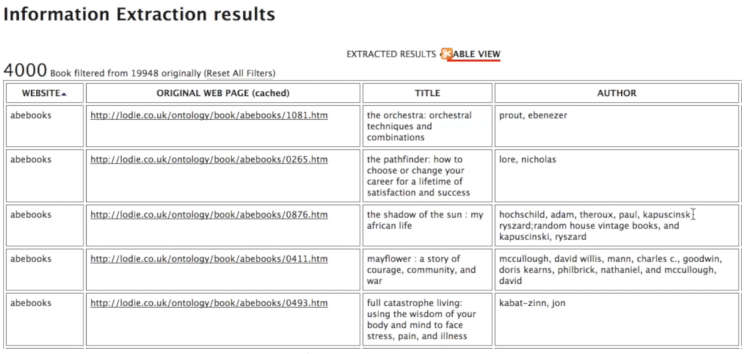

Fig. 4: Exploration of results produced by the IE method

References

1. Kushmerick, N.: Wrapper Induction for information Extraction. In: IJCAI97. (1997) 729–735
2. Gentile, A.L., Zhang, Z., Augenstein, I., Ciravegna, F.: Unsupervised wrapper induction using linked data. In: Proc. of the seventh international conference on Knowledge capture. K-CAP '13, New York, NY, USA, ACM (2013) 41–48
3. Gentile, A.L., Zhang, Z., Ciravegna, F.: Self training wrapper induction with linked data. In: Proceedings of the 17th International Conference on Text, Speech and Dialogue (TSD 2014). (2014) 295–302
4. Ciravegna, F., Gentile, A.L., Zhang, Z.: Lodie: Linked open data for web-scale information extraction. In: SWAIE. (2012) 11–22
5. Muslea, I., Minton, S., Knoblock, C.: Hierarchical wrapper induction for semistructured information sources. Autonomous Agents and Multi-Agent Systems (2001) 1–28
6. Soderland, S.: Learning information extraction rules for semi-structured and free text. Mach. Learn. **34**(1-3) (February 1999) 233–272
7. Muslea, I., Minton, S., Knoblock, C.: Active Learning with Strong and Weak Views: A Case Study on Wrapper Induction. IJCAI'03 8th international joint conference on Artificial intelligence (2003) 415–420
8. Dalvi, N., Kumar, R., Soliman, M.: Automatic wrappers for large scale web extraction. Proc. of the VLDB Endowment **4**(4) (2011) 219–230
9. Wong, T., Lam, W.: Learning to adapt web information extraction knowledge and discovering new attributes via a Bayesian approach. Knowledge and Data Engineering, IEEE **22**(4) (2010) 523–536
10. Hao, Q., Cai, R., Pang, Y., Zhang, L.: From One Tree to a Forest : a Unified Solution for Structured Web Data Extraction. In: SIGIR 2011. (2011) 775–784
11. Crescenzi, V., Mecca, G.: Automatic information extraction from large websites. Journal of the ACM **51**(5) (September 2004) 731–779
12. Arasu, A., Garcia-Molina, H.: Extracting structured data from web pages. In: Proc. of the 2003 ACM SIGMOD international conference on Management of data, ACM (2003) 337–348
13. Mazumdar, S., Petrelli, D., Elbedweihy, K., Lanfranchi, V., Ciravegna, F.: Affective graphs: The visual appeal of linked data. Semantic Web–Interoperability, Usability, Applicability. IOS Press (to appear, 2014) (2013)