Knowledge extraction from webpages

Sylvain Tenier, Amedeo Napoli, Xavier Polanco, Yannick Toussaint

Knowledge extraction from webpages

Sylvain Tenier[1,2], Amedeo Napoli[2], Xavier Polanco[1], and Yannick Toussaint[2]

[1] Institut National de l'Information Scientifique et Technique
54514 Vandoeuvre-ls-Nancy, France
{polanco,tenier}@inist.fr,
http://www.inist.fr/uri/accueil.htm
[2] Laboratoire Lorrain de Recherche en Informatique et ses Applications
BP 239, 54506 Vandoeuvre ls Nancy Cedex, France
napoli,toussaint,tenier}@loria.fr
http://www.loria.fr/equipes/orpailleur

Abstract. This article presents a system to extract Knowledge from webpages by producing semantic annotations. taking into account semantic information from the domain to annotate an element in a webpage implies solving two problems : (1) identifying the syntactic structure of this element in the webpage and (2) identifying the most specific concept (in terms of subsumption) of the ontology that will be used to annotate this element. Our approach relies on a wrapper-based machine learning algorithm combined with reasoning making use of the formal structure of the ontology.

1 Context of the research

Our system aims at using information provided by research teams on their website to generate knowledge about the European Research Community. In order to make this information machine-processable, a formal representation of the content of the webpages is needed, encoded with a well-defined syntax and semantics. This is the purpose of semantic annotation [1]. The system is provided with :

- an ontology which represents the concepts of a domain and their relationships. The ontology, implemented in the Web Ontology Language (OWL), is based on Description Logics (DL) and thus reasoning mechanisms, like classification and subsumption, are provided [2],
- webpages from which data are extracted according to the ontology.

For each data in the document, the systems generates an individual with the concept and roles it instantiates. Each individual is added to a Knowledge Base (KB). Two main tasks are dealt with: the first is about locating each data in the provided documents and extracting it to generate a "raw" individual which may not be specific enough. It is followed by a reasoning task which infers the most specific concept the individual is an instance of.

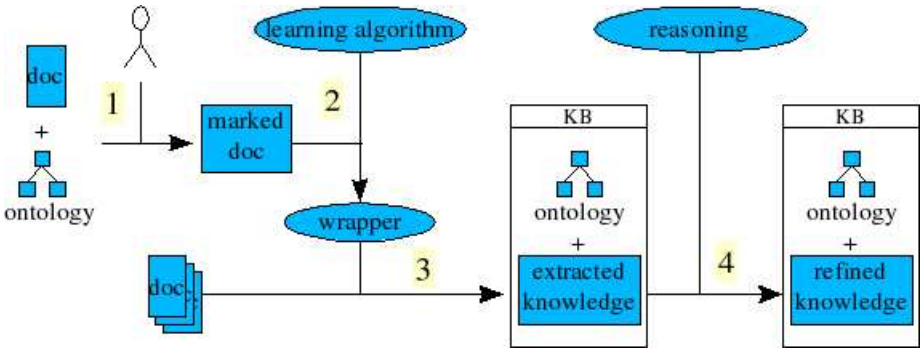

Fig. 1. overview of the approach

2 Extracting knowledge

2.1 Manual annotation

The system is based on machine learning techniques (fig. 1). It learns from some examples the data to be extracted. Here, the examples are webpages annotated by hand according to an ontology. This task is performed in a dedicated environment in which a user is presented a page to annotate and the concepts of the ontology in a browser-like interface. The user then annotates a few occurences of the concepts of the ontology he wants the system to identify. The number of annotations needed depends on the regularity of the page. If the data is strongly structured, like tables, only two or three examples are needed. For example, to extract knowledge about research teams, the *SWRC* ontology (fig. 2) is loaded along with a page presenting the persons working in a team and the projects they are involved in. The user then annotates some data that are instances of the concept of *Person* and some instantiating the concept of *Project*. The output is a marked document in which the annotations are embedded.

2.2 Wrapper induction using the tree structure of the page

The learning algorithm is derived from Kushmerick's work on wrapper induction, which identifies classes of wrappers than can be learnt using a deterministic algorithm with low complexity [3]. A wrapper is a procedure using the syntactic regularities of a document to extract data. This is particularly suited to semi-structured documents like webpages. We have adapted Kushmerick's work to make use of the tree structure of a webpage provided by the w3c's Document Object Model (DOM). This model defines a path leading to each data in the document. The marked page is taken as input and the similarities between paths leading to data which are instances of the same concept are learnt. The output is a wrapper which is applied to pages in which the tree structure is similar to the example page in order to extract the data and their relationships.

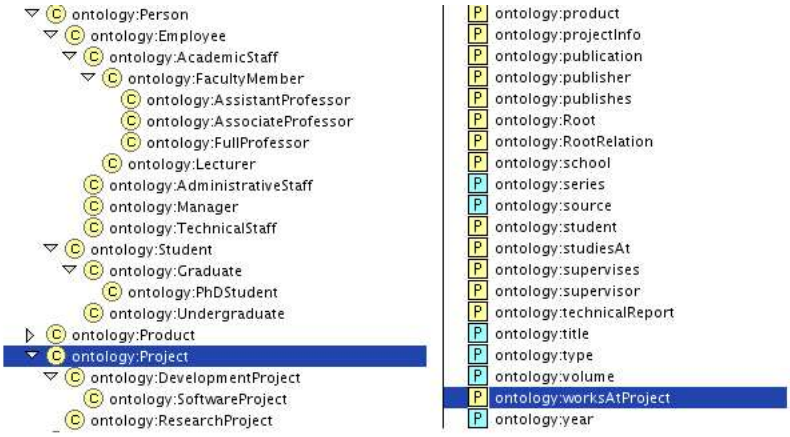

Fig. 2. concepts and properties of the SWRC ontology

2.3 Adding knowledge to the KB

In step 3, the relevant wrappers are applied to the documents to extract data. For each extracted data, an instance of the concept it belongs to is added to the KB together with its relationships with other data. The resulting KB is a graph implemented using the Resource Description Framework (RDF) which connects each individual to the ontology and the individuals it is related to. At that point, one problem is that the extracted knowledge is not specific enough. The reason is that a wrapper must be as generic as possible in order to extract all the relevant data in a document. Therefore, it is induced using the most general concept available. For example, to extract people from a research team according to the *SWRC* ontology, the wrapper will be designed to recognize any instance of the concept *Person* instead of more specific concepts like *AcademicStaff* or *Student*. The second limitation is that the semantics of the relationships is unknown, since they are extracted using syntactic properties (for instance, two data are related if they have a common parent node).

3 Refining the knowledge

With an ontology implemented in the Knowledge Representation language OWL, reasoning mechanisms are provided. One of them is *instantiation*, which given an individual and a set of concepts finds the most specific concept the individual belongs to, with respect to the subsumption ordering. For example, in the *SWRC* ontology, two concepts are subsumed by the *Person* concept and have a role whose filler must be an instance of the *Project* concept. Therefore, the individuals in the KB that are instances of a *Person* and have a relationship to a *Project* must be either instances of *AcademicStaff* or *PhdStudent* and the relationship is identified as being the *Project* role. The new knowledge is inserted into the KB.

4 Conclusion

We have presented a system that integrates semantics in the annotation generation process by making use of the reasoning mechanisms provided by the ontology according to which webpages are annotated. This requires not only concept instances but also role instances to be extracted. Initial works on webpage annotation such as Annotea [4] aimed at enabling collaborative work between people. The need for annotations machines could understand and reason with led to systems, such as S-CREAM [5], producing semantic annotations according to Description Logics based ontologies. Since manual annotation is a tedious and error-prone task, machine learning system have been proposed; S-CREAM and MnM [6] implement Amilcare [7], a semi-automatic tool that produces extraction rules from a corpus to generate concept instances; however, role instances are not dealt with. Recently, fully automatic systems like Amardillo [8] or C-Pankow [9] have been presented that make use of the redundancy of data on the Web. Our system relies on the hypothesis of a of mapping between the syntax and the semantics of a webpage. Since its efficiency depends on the presence of regularities in the structure, pages from research team websites are well suited for this task.

References

1. Kiryakov, A., Popov, B., Ognyanoff, D., Manov, D., Kirilov, A., Goranov, M.: Semantic annotation, indexing, and retrieval. In: International Semantic Web Conference. (2003) 484–499
2. Horrocks, I., Sattler, U., Tobies, S.: Practical reasoning for very expressive description logics. CoRR **cs.LO/0005013** (2000)
3. Kushmerick, N., Weld, D.S., Doorenbos, R.B.: Wrapper induction for information extraction. In: IJCAI (1). (1997) 729–737
4. Kahan, J., Koivunen, M.R.: Annotea: an open rdf infrastructure for shared web annotations. In: WWW '01: Proceedings of the 10th international conference on World Wide Web, New York, NY, USA, ACM Press (2001) 623–632
5. Handschuh, S., Staab, S., Ciravegna, F.: S-cream-semi-automatic creation of metadata. Proc. of the European Conference on Knowledge Acquisition and Management (2002) Springer Verlag (submitted version).
6. Vargas-Vera, M., Motta, E., Domingue, J., Lanzoni, M., Stutt, A., Ciravegna, F.: Mnm: Ontology driven semi-automatic and automatic support for semantic markup. In: EKAW. (2002) 379–391
7. Ciravegna, F., Dingli, A., Wilks, Y., Petrelli, D.: Adaptive information extraction for document annotation in amilcare. In: SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, New York, NY, USA, ACM Press (2002) 451–451
8. Ciravegna, F., Chapman, S., Dingli, A., Wilks, Y.: Learning to harvest information for the semantic web. In: ESWS. (2004) 312–326
9. Cimiano, P., Ladwig, G., Staab, S.: Gimme' the context: context-driven automatic semantic annotation with c-pankow. In: WWW '05: Proceedings of the 14th international conference on World Wide Web, New York, NY, USA, ACM Press (2005) 332–341