Towards Combining Machine Learning with Attribute Exploration for Ontology Refinement

Jedrzej Potoniec[1], Sebastian Rudolph[2], and Agnieszka Ławrynowicz[1]

[1] Institute of Computing Science, Poznan University of Technology, Poland
{jpotoniec,alawrynowicz}@cs.put.poznan.pl
[2] Technische Universität Dresden, Germany sebastian.rudolph@tu-dresden.de

Abstract. We propose a new method for knowledge acquisition and ontology refinement for the Semantic Web utilizing Linked Data available through remote SPARQL endpoints. This method is based on combination of the attribute exploration algorithm from formal concept analysis and the active learning approach from machine learning.

1 Introduction

Knowledge acquisition is a process of capturing knowledge, typically from a human expert, and thus it concerns all systems and environments where that kind of knowledge is required. It is also said to be major bottleneck in development of intelligent systems due to its difficulty and time requirements. Of course the Semantic Web, as an area concerned with structured and precise representation of information, has to deal with exactly the same issue.

Since the early days of the Semantic Web, building ontologies has been a difficult and laborious task. Frequently people trying to express complex knowledge do not know how to perform this task properly. Mistakes come from difficulty in understanding the complex logic formalism supporting OWL.

Frequently an ontology engineer would start collecting vocabulary and requirements for an ontology, structuralize the vocabulary and later specify more complex dependencies [6]. We propose a solution to support knowledge acquisition for ontology construction. Especially we address the last part of the process, where some basic knowledge is already gathered and more complex dependencies are to be specified. We aim to answer the question *how to extend an ontology with meaningful, valid and non-trivial axioms taking into consideration available data and user workload*?

2 Related work

For knowledge acquisition for ontology development many approaches have been proposed so far. The most basic ones are ontology editors supporting ontology development, such as *Protégé*[3]. In addition to that, there are methodologies helpful in ontologies development, such as the one proposed in *NeOn* [6].

[3] http://protege.stanford.edu/

In [1,5] applications of attribute exploration algorithm from formal concept analysis to ontology development have been proposed. [1] describes how to discover subsumptions between conjunction of classes and [5] extends it to properties' domains and ranges.

In [3] the idea of learning ontologies purely from Linked Data by means of discovering association rules is presented. [2] presents a methodology for manually building and populating domain ontologies from Linked Data.

3 Approach

The proposed approach is to support the user during attribute exploration by means of machine learning (ML). The ML algorithm's task is to answer simple, non-interesting questions posed by the attribute exploration algorithm and leave for the user only these questions which are non-trivial to answer.

Input of our proposed algorithm is an ontology $\mathcal{O}$, a partial context derived from it, and two thresholds θ_a and θ_r. They are, respectively, thresholds for accepting and rejecting an implication and have to be manually chosen w.r.t. the used ML algorithm. Result of the algorithm is a set of probably valid implications, which can be transformed into subsumptions for extending the ontology. A detailed description of the algorithm is presented below. For sake of clarity its description treats the attribute exploration algorithm as a black box, which provides the next implication to consider.

1. Generate implication $L \to R$ by means of the attribute exploration algorithm.
2. For every $r \in R$, do the following sequence of steps:
   (a) If $L \to \{r\}$ is already refuted by some of the known individuals, go to the next r.
   (b) If $\mathcal{O} \models \bigsqcap L \sqsubseteq r$, remember implication $L \to \{r\}$ as a valid one and go to the next r.
   (c) Compute probabilities of acceptance p_a and rejection p_r of the implication $L \to \{r\}$ with the ML algorithm. Note that $p_a + p_r = 1$.
   (d) If $p_a \geq \theta_a$, remember the implication $L \to \{r\}$ as a valid one and go to the next r.
   (e) If $p_r \geq \theta_r$, go to the step 2i.
   (f) Ask user if implication $L \to \{r\}$ is valid.
   (g) Add considered implication with user's answer to a set of learning examples for the ML algorithm.
   (h) If the implication is valid, remember it as a valid one and go to the next r.
   (i) Otherwise, extend the partial context with a counterexample either provided by user or auto-generated.

The purpose of iteration through the set of conclusions R in the algorithm is twofold. We believe that this way user can more easily decide if the presented

implication is valid or not, because she does not have to consider complex relation between two conjunctions of attributes.

The other thing is that this way automated generation of counterexamples provides more concrete results. For an arbitrary implication $L \to R$ a counterexample can be generated and said to have all attributes from L and to not have at least one attribute from R. This is not in line with the method of partial context induction, as it is unclear which exactly attribute from R the counterexample does not have. Because of that partial context can not reflect knowledge base accurately anymore, and the attribute exploration algorithm can start to generate invalid implications. If the implication has a single attribute in its right-hand side, it is clear which attribute the counterexample does not have.

3.1 Application of machine learning

The task which ML algorithm is to solve can be seen as a kind of active learning with a binary classification. Every implication is classified as *valid* or *invalid* and if the algorithm is unsure, the user is asked.

One should note that not every classifier generates reasonable probabilities. For example, rule-based or tree-based systems usually are not suitable for that purpose. Problem of generating probabilities can be also seen as a regression problem.

Moreover costs of both types of mistakes are different and distribution of learning examples can be heavily imbalanced, i.e. implications with one decision may appear much often than with other decision. To reflect these fact a classifier suitable for cost-sensitive learning is required.

To apply machine learning techniques, a way to transform implications to feature vectors is required. We apply three approaches to this problem. First of all, a single purely syntactic measure is used: the number of attributes in the left-hand side divided by the number of all attributes. Secondly, there are features made of values of measures typical for association rules mining. Their computation is based on features of individuals in the ontology. Following the naming convention from [4], we use *coverage, prevalence, support, recall* and *lift*.

Finally, we use a mapping from the set of the attributes to Linked Data in order to obtain the number of objects in an RDF repository supporting an implication or its parts. Every attribute is mapped to a SPARQL graph pattern with a single featured variable denoting the object identifier. Following the same name convention from [4], *coverage, prevalence, support, recall* and *confidence* are used. All of these features can be computed using only SPARQL `COUNT DISTINCT` expressions and basic graph patterns and thus they maintain relatively low complexity and are suitable to use with remote SPARQL endpoints.

Such a feature vector is later labeled with the classifier mentioned above and given answer (valid/invalid/unsure) is used to either refine the ontology or ask the user. If the user is asked, her answer is then used as a correct label for the feature vector and the classifier is relearned.

4 Conclusions and future work

As we are proposing a method which is to make development and refinement of domain-specific ontologies easier, our main goal for evaluation is to validate its practical usability. We plan to apply our method to a selection of domain-specific ontologies concerning some knowledge of general type such as literature, music and movies. We plan to use a crowdsourcing service to validate our hypotheses. We hope that with ontologies with a theme being general enough and additional information available in the Internet, the crowd will be able to validate our decisions about implications and Linked Data mappings.

We believe that our approach is promising and will be able to help ontology engineers in the process of ontology refinement. We are combining three technologies very suitable for this kind of a task. First of all, the attribute exploration algorithm that has been developed especially for discovering additional relations between attributes. Moreover, Linked Data is supposed to describe parts of the world. Obviously, this description can not be assumed to be neither accurate nor complete, yet it should be sufficient to support the user in a process of ontology refinement. Finally, the whole purpose of machine learning algorithms is to adapt themselves, and thus they are suitable to replace the user in uniform, repeatable tasks.

Acknowledgement. Jedrzej Potoniec and Agnieszka Ławrynowicz acknowledge support from the PARENT-BRIDGE program of Foundation for Polish Science, cofinanced from European Union, Regional Development Fund (Grant No POMOST/2013-7/8 *LeoLOD – Learning and Evolving Ontologies from Linked Open Data*).

References

1. Baader, F., Ganter, B., et al.: Completing description logic knowledge bases using formal concept analysis. In: Proc. of IJCAI 2007. pp. 230–235. AAAI Press (2007)
2. Dastgheib, S., Mesbah, A., Kochut, K.: mOntage: Building Domain Ontologies from Linked Open Data. In: IEEE Seventh International Conference on Semantic Computing (ICSC). pp. 70–77. IEEE (2013)
3. Fleischhacker, D., Völker, J.: Inductive learning of disjointness axioms. In: Meersman, R., Dillon, T., et al. (eds.) On the Move to Meaningful Internet Systems: OTM 2011, LNCS, vol. 7045, pp. 680–697. Springer Berlin Heidelberg (2011)
4. Le Bras, Y., Lenca, P., Lallich, S.: Optimonotone measures for optimal rule discovery. Computational Intelligence 28(4), 475–504 (2012)
5. Rudolph, S.: Acquiring generalized domain-range restrictions. In: Medina, R., Obiedkov, S. (eds.) Formal Concept Analysis, LNCS, vol. 4933, pp. 32–45. Springer Berlin Heidelberg (2008)
6. Suárez-Figueroa, M.C., Gómez-Pérez, A., Fernández-López, M.: The NeOn Methodology for Ontology Engineering. In: Suárez-Figueroa, M.C., Gómez-Pérez, A., et al. (eds.) Ontology Engineering in a Networked World, pp. 9–34. Springer Berlin Heidelberg (2012)