Open Research Online

Citation

URL

https://oro.open.ac.uk/23496/

License

Policy

Versions

SCARLET: SemantiC relAtion discoveRy by harvesting onLinE onTologies

Marta Sabou, Mathieu d'Aquin, and Enrico Motta

Knowledge Media Institute (KMi), The Open University, Milton Keynes,
{r.m.sabou, m.daquin, e.motta}@open.ac.uk

Abstract. We present a demo of SCARLET, a technique for discovering relations between two concepts by *harvesting the Semantic Web*, i.e., *automatically* finding and exploring *multiple* and *heterogeneous* online ontologies. While we have primarily used SCARLET's relation discovery functionality to support ontology matching and enrichment tasks, it is also available as a stand alone component that can potentially be integrated in a wide range of applications. This demo will focus on presenting SCARLET's functionality and its different parametric settings that can influence the trade-off between its accuracy and time performance.

1 Introduction

A novel trend of harvesting the Semantic Web, i.e., *automatically* finding and exploring *multiple* and *heterogeneous* online knowledge sources, has been favored by the recent growth of online semantic data and the increased interest in building gateways that allow quick exploration of this data[1] (e.g., Watson [4]). For example, Alani proposes a method for ontology learning that relies on cutting and pasting ontology modules from online ontologies relevant to keywords from a user query [1]. Then, in [5] the authors describe a multi-ontology based method that exploits the Semantic Web rather than WordNet to disambiguate the senses of keywords that are given as a query to a search engine (e.g., *star* is used in its sense of celestial body in [*astronomy, start, planet*]).

SCARLET [2] follows this paradigm of automatically selecting and exploring online ontologies *to discover relations between two given concepts*. For example, when relating two concepts labeled *Researcher* and *AcademicStaff*, SCARLET 1) identifies (at run-time) online ontologies that can provide information about how these two concepts inter-relate and then 2) combines this information to infer their relation. We describe two increasingly sophisticated strategies to discover and exploit online ontologies for relation discovery. The first strategy derives a relation between two concepts if this relation is defined within a single online ontology, e.g., stating that *Researcher* $\sqsubseteq$ *AcademicStaff* (Section 2.1). The second strategy (Section 2.2) addresses those cases when no single online ontology states

---

[1] http://esw.w3.org/topic/TaskForces/CommunityProjects/LinkingOpenData/
SemanticWebSearchEngines
[2] http://scarlet.open.ac.uk/

the relation between the two concepts by combining relevant information which is spread over two or more ontologies (e.g., that $Researcher \sqsubseteq ResearchStaff$ in one ontology and that $ResearchStaff \sqsubseteq AcademicStaff$ in another). The DBPedia Relation Finder [6] provides a similar functionality (it discovers connections between two objects in the DBPedia data set), but it relies on semantically weaker methods (e.g., graph clustering instead of reasoning).

SCARLET originates from earlier work in the field of ontology matching, from the design of a matcher that exploits the entire Semantic Web as a source of background knowledge [7]. In essence, this matcher discovers semantic relations (mappings) between the elements of two ontologies by using the methods described above. A large-scale evaluation of this matcher lead to precision values of over 70% [8]. SCARLET's relation discovery functionality has also been used to semantically enrich folksonomy tagsets [2]. Given a set of implicitly related tags, we used SCARLET to identify relations between these tags and then merged them into a new knowledge structure (ontology).

While we have used SCARLET to support two different tasks, we also provide it as a stand-alone component as its functionality could be useful when integrated in a variety of other tools. To cater for applications with different needs, SCARLET can be used with a variety of parametric settings that regulate the trade-off between its accuracy and time performance. In this demo we provide an insight in the internal working of SCARLET and its different settings.

2   SCARLET: Technology Overview

In this section we present an overview of SCARLET's two strategies (as described in [7]) and its main parameters. The parameters can be fine-tuned from the command line, and soon through a Web based interface. Each strategy is presented as a procedure that takes two candidate concepts (denoted as A and B) as an input and returns the discovered relation between them. The corresponding concepts to A and B in an online ontology O_i are A'_i and B'_i ("anchor terms"). We rely on the description logic syntax for semantic relations occurring between concepts in an online ontology O_i (e.g., $A'_i \sqsubseteq B'_i$) and on a C-OWL like notation [3] for the returned relations(e.g., $A \xrightarrow{\sqsubseteq} B$). Note that the current version of SCARLET only explores taxonomic and disjoint relations.

2.1   Strategy S1: Relation Discovery Within One Ontology

Strategy S1 consists of finding ontologies containing concepts similar with the candidate concepts (e.g., by relying on Watson) and then deriving a relation from their relations in the selected ontologies. Figure 1 (a) illustrates this strategy with an example where three ontologies are discovered (O_1, O_2, O_3) containing the concepts `A'` and `B'` corresponding to `A` and `B`. The first ontology contains no relation between the anchor concepts, while the other two ontologies declare a subsumption relation. The concrete steps of this strategy are:

1. Anchor A and B to corresponding concepts A' and B' in online ontologies;
2. Select ontologies containing A' and B';
3. For a given ontology (O_i) apply the following rules:
   – if $A'_i \equiv B'_i$ then derive $A \xrightarrow{\equiv} B$;
   – if $A'_i \sqsubseteq B'_i$ then derive $A \xrightarrow{\sqsubseteq} B$;
   – if $A'_i \sqsupseteq B'_i$ then derive $A \xrightarrow{\sqsupseteq} B$;
   – if $A'_i \perp B'_i$ then derive $A \xrightarrow{\perp} B$;
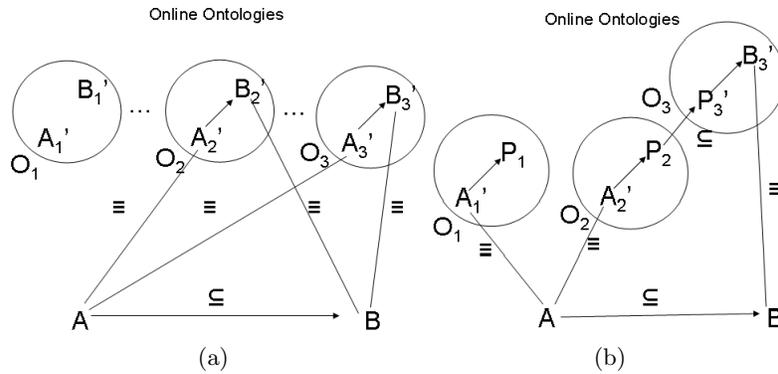4. Combine all relations derived from the considered ontologies.

Fig. 1. Relation discovery (a) within one ontology (S1) and (b) across ontologies (S2).

For example, when matching two concepts labeled *Drinking Water* and *tap_water*, appropriate anchor terms are discovered in the TAP ontology and the following subsumption chain in the external ontology is used to deduce the relation: $DrinkingWater \sqsubseteq FlatDrinkingWater \sqsubseteq TapWater$.

2.2   Strategy S2: Cross-Ontology Relation Discovery

The previous strategy assumes that a relation between the candidate concepts can be discovered in a single ontology. However, some relations could be distributed over several ontologies. Therefore, if no ontology is found that relates both candidate concepts, then the relation should be derived from two (or more) ontologies. In this strategy, relation discovery is a recursive task where two concepts can be matched because the concepts they relate to in some ontologies are themselves matched. Figure 1 (b) illustrates this strategy where no ontology is available that contains anchor terms for both A and B, but where one of the parents (P_2) of the anchor term A'_2 can be matched to B in the context of a third ontology (O_3). For example, a relation between *Cabbage* and *Meat* can be

derived by taking into account that $Cabbage \sqsubseteq Vegetable$[3] and then discovering that $Vegetable \perp Meat$[4] through another discovery step. The concrete steps are:

1. Anchor A and B to corresponding concepts A' and B' in online ontologies;
2. If no ontologies are found that contain both A' and B' then select all ontologies containing A';
3. For a given ontology O_i apply the following rules:
   (a) for each P_i such that $A'_i \sqsubseteq P_i$, search for relations between P_i and B;
   (b) for each C_i such that $A'_i \sqsupseteq C_i$, search for relations between C_i and B;
   (c) derive relations using the following rules:
       – (r1) if $A'_i \sqsubseteq P_i$ and $P_i \xrightarrow{\sqsubseteq} B$ then $A \xrightarrow{\sqsubseteq} B$
       – (r2) if $A'_i \sqsubseteq P_i$ and $P_i \xrightarrow{\equiv} B$ then $A \xrightarrow{\sqsubseteq} B$
       – (r3) if $A'_i \sqsubseteq P_i$ and $P_i \xrightarrow{\perp} B$ then $A \xrightarrow{\perp} B$
       – (r4) if $A'_i \sqsupseteq C_i$ and $C_i \xrightarrow{\sqsupseteq} B$ then $A \xrightarrow{\sqsupseteq} B$
       – (r5) if $A'_i \sqsupseteq C_i$ and $C_i \xrightarrow{\equiv} B$ then $A \xrightarrow{\sqsupseteq} B$
4. Combine all relations derived from the considered ontologies.

2.3   Main Parameters

There are several parameters that can influence the way SCARLET works.

Ontology gateway Currently, SCARLET can rely on either Swoogle or Watson to access online available ontologies.

Strategy SCARLET can function according to two different strategies: S1, when the relation is derived from within a single ontology and S2, when information spread over several ontologies is combined to derive a relation.

Number of derived relations The anchoring step of both strategies (step 1), identifies all the ontologies that possibly contain a relation between the input concepts. An important decision is the number of ontologies used to derive relations. On the one hand, using a single ontology is the easiest way to deal with the multiple returned ontologies but it assumes that the discovered relation can be trusted and there is no need to inspect the other ontologies as well. On the other hand, using a subset (or all) of the returned ontologies is computationally more expensive but it has a higher accuracy by taking into account all the information that can be possibly derived. In these cases a relation is derived from each ontology and then these are combined into a final relation (using a method selected by the next parameter).

Method for combining multiple relations If all possible relations between the terms are derived, then it is important to specify what will actually be returned by SCARLET. By default, SCARLET returns all the derived relations. Another option is to return a relation only if all the derived relations were the same (i.e., all ontologies stated the same relation). A final possibility is to return the relation that was most frequently derived, i.e., on which most sources agree.

---

[3] `http://139.91.183.30:9090/RDF/VRP/Examples/tap.rdf`
[4] `http://www.co-ode.org/resources/ontologies/Pizzademostep1.owl`

S2: Depth of hierarchy considered The number of parent/child classes in the anchor ontology used in S2 have an influence on the complexity of the algorithm. In the simplest case, only the direct parents/children are considered. In the most complex case, S2 will investigate all parents/children. The larger the considered depth, the longer the processing times and (possibly) higher the number of discovered relations.

3   Demonstration Plan

The demo will focus on showcasing the functionality of SCARLET as well as the use of its various parameters. Additional information will be provided about the paradigm on which this technique relies, the case studies in which it was used and its internal functioning.

We expect that several aspects of this demo will be of interest to the visitors. First, the visitor will learn about the new paradigm of harvesting the Semantic Web and experience through SCARLET one of its concrete implementations. Second, he will get to know the functionality of SCARLET and the types of tasks in which it has already been used. This information could help him to understand whether SCARLET could be useful in his own work. Finally, visitors interested in the internal working of SCARLET, will be explained how the software works and how it can be fine-tuned through its various parameters. Those visitors that intend to use this functionality in their own work could already determine what settings suit their needs best.

References

1. H. Alani. Position Paper: Ontology Construction from Online Ontologies. In *Proc. of WWW*, 2006.
2. S. Angeletou, M. Sabou, L. Specia, and E. Motta. Bridging the Gap Between Folksonomies and the Semantic Web: An Experience Report. In *Proc. of the ESWC Workshop on Bridging the Gap between Semantic Web and Web 2.0*, 2007.
3. P. Bouquet, F. Giunchiglia, F. van Harmelen, L. Serafini, and H. Stuckenschmidt. Contextualizing ontologies. *Journal of Web Semantics*, 1(4):24, 2004.
4. M. d'Aquin, C. Baldassarre, L. Gridinoc, M. Sabou, S. Angeletou, and E. Motta. Watson: Supporting Next Generation Semantic Web Applications. In *Proc. of WWW/Internet conference*, Vila Real, Spain, 2007.
5. J. Gracia, R. Trillo, M. Espinoza, and E. Mena. Querying the Web: A Multiontology Disambiguation Method. In *Proc. of ICWE*, 2006.
6. J. Lehmann, J. Schuppel, and S. Auer. Discovering Unknown Connections - the DBpedia Relationship Finder. In *Proc. of CSSW*, 2007.
7. M. Sabou, M. d'Aquin, and E. Motta. Using the Semantic Web as Background Knowledge for Ontology Mapping. In *Proc. of the Ontology Matching WS*, 2006.
8. M. Sabou, J. Gracia, S. Angeletou, M. d'Aquin, and E. Motta. Evaluating the Semantic Web: A Task-based Approach. In *Proc. of ASWC/ISWC*, 2007.