FirstOnt: Automatic Construction of Ontologies out of Multiple Ontological Resources

Carlos Bobed, Eduardo Mena, and Raquel Trillo

IIS Department
University of Zaragoza
50018 Zaragoza, Spain,
{cbobed,emena,raqueltl}@unizar.es

Abstract. Nowadays, a great amount of the contents of the WWW are still mainly only human-oriented. To progressively move into the Semantic Web, the adoption of the use of ontologies is a milestone. However, their elaboration from scratch is quite expensive, which could prevent non-experts from using them in their applications. Due to this reason, different approaches for ontology reusing and engineering have been proposed, but they require domain experts and knowledge engineers. Besides, finding an ontology (if it exists) that fits a specific domain can be a tedious and frustrating experience.

In this paper, we propose an approach that, taking as starting point a set of user keywords that define the domain of interest, automatically integrates an ontology that comprises information extracted from different relevant resources. This ontology can be directly used as an initial version for a progressive refinement or as a final resource for non-expert users helping to spread the use of ontologies. This way, our proposal succeeds in alleviating the development efforts to create an ontology and facilitate this task to non-expert users. Moreover, it encourages ontology reuse and to reach a consensus when using these ontologies, for instance, to publish Linked Data.

Keywords: Ontologies, Ontological Engineering, Semantic Web

1 Introduction

The usefulness of the Semantic Web is beyond doubt, enabling computers to know which meaning is behind the different Web resources. However, there is still a milestone to manage the successful adoption of the Semantic Web: to make the access and reuse of ontologies, its main tools, easy to final users. Analyzing the fast spread of HTML and the adoption of the WWW as part of users' life, one of the most important factors that helped this to happen was the ease of creating Web pages by reusing chunks of another ones. This lessened the need of mastering HTML to have a Web page published with world wide access. Thus, for a successful adoption of the Semantic Web, apart from showing the users the benefits of adding semantics to their resources, we have to minimize their efforts needed to do so.

From the point of view of software engineering, reuse is desirable and its importance arises even more remarkably in this field. An ontology is defined as a shared specification of a conceptualization [12], so, the more users using the same ontologies, the higher level of global agreement can be achieved. This agrement would help to develop better ontology-guided search engines as relatively few ontologies would be accepted by everyone. Moreover, this is consistent with the Semantic Web ontologies envisioned in [13]: "*instead of a few large, complex, consistent ontologies that great users share, there may be a great number of small ontological components consisting largely of pointers to each other*". However, reusing ontologies is not an easy task and has been a subject of study since the very beginning of the Semantic Web [18, 19]. In [19], the authors pointed out the two different reuse processes: *merge* and *integration*. Merging is the process of building an ontology in one subject reusing two or more different ontologies on that subject, while integration is the process of building an ontology in one subject reusing one or more ontologies *on different subjects*. We advocate for both processes to be performed to obtain proper results.

In [1], the author envisioned a system that would build ontologies semiautomatically from available online ontologies by using segmentation, mapping and merging techniques. In this paper, we propose an approach to make it become real. Taking as input a list of keywords that represent the main terms that are conforming the desired ontology, our system integrates automatically an ontology reusing existing ontological information. The first step is to disambiguate the meaning behind the user's keywords [21], which gives us the possible interpretations of the keywords having into account their context. This establishes the meaning of each keyword and associate to it a multi-ontology sourced ontological context. Then, our system integrates these contexts in one ontology, while registering the sources referenced in them. To further improve the quality of the resulting ontology in terms of relationships and definitions, our system uses several techniques to obtain more ontological information from the source ontologies. Finally, with the help of a Description Logics reasoner [2], our system removes redundancies and detects possible inconsistencies. The resulting ontology can be used by knowledge engineers as a starting point for later refinements (allowing them to automatize the discovering and reuse of sources) or even by final users to add semantics to their own resources without further refinement.

The rest of the paper is organized as follows. First, in Section 2, we give an overview of the integration process. In Section 3, we explain the keyword disambiguation techniques applied by our system. In Section 4, we detail how our system exploits the sources of information discovered during the disambiguation process, and how it uses them to integrate the resulting ontology. Section 5 presents an application scenario in which the use of our approach can be very useful. Related works are presented in Section 6. Finally, conclusions and future work are presented in Section 7.

2 Overview of the Process

The main goal of our system is to facilitate the construction of ontologies. To make it easier the capture of requirements of the ontology to the user, the system takes as input a list of keywords that describes his/her needs. As can be seen in Figure 1, the main process consists of two steps: *Keyword Senses Discovery*, and *Ontology Integration*.

The aim of the first step is to obtain the exact meaning of the keywords input by the user. Our discovery and disambiguation process is based on two semantic measurements defined in previous work: the *Ontology-based synonym probability*, and a *Web-based relatedness measure* [21]. The system uses the Ontology-based synonym probability to dynamically create a set of senses for each keyword; and, after that, it employs the Web-based relatedness measures to rank the most proper sense of each keyword taking into account their context (i.e. the whole set of keywords provided by the user). In Section 3, we briefly describe the steps to achieve this goal.

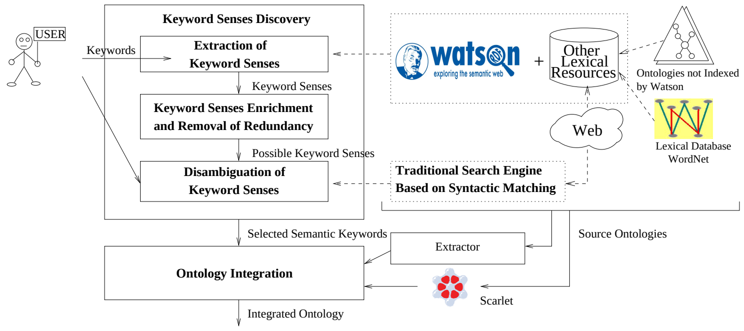

USER
Keywords
Keyword Senses Discovery
Extraction of Keyword Senses
Keyword Senses
Keyword Senses Enrichment and Removal of Redundancy
Possible Keyword Senses
Disambiguation of Keyword Senses
watson exploring the semantic web
+
Other Lexical Resources
Ontologies not Indexed by Watson
Lexical Database WordNet
Web
Traditional Search Engine Based on Syntactic Matching
Selected Semantic Keywords
Extractor
Source Ontologies
Ontology Integration
Scarlet
Integrated Ontology

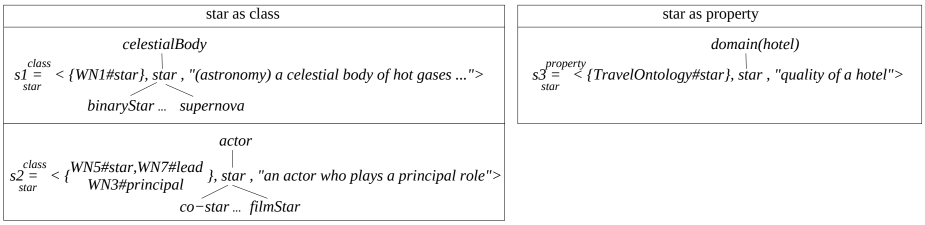
Fig. 1. Overview of the process

The second step takes as input a set of *semantic keywords*, which are defined as the pair formed by a keyword and its attached sense. Each sense stores the ontological information that has been used during the keyword disambiguation process. This information is integrated into one ontology maintaining the original references and it is used to obtain further information from the original sources. In Section 4, we detail how this information is integrated and used to obtain a richer final ontology.

3 Discovery of Keyword Senses

As stated in the previous section, the first step that our system performs is the discovery of the semantics that exists behind the user keywords. This discovery is done by taking into account the individual possible semantics of each keywords as well as the possible semantics of its context (the rest of keywords). In particular, this process is divided into three substeps (see Figure 1):

- Extraction of Keyword Senses: The system extracts out the possible meanings of each keyword from a dynamic pool of ontologies (in particular, it queries Watson [8], WordNet [15] and other ontology repositories to find ontological terms that syntactically match the keywords - or one of their synonyms). The system builds a sense for each matching obtained and then the extracted senses are semantically enriched with the ontological terms of their synonyms by also searching in the ontology pool. The result is a list of candidate keyword senses for each user keyword. In Figure 2, three possible senses retrieved for user keyword *star* are shown.



star as class
celestialBody
s1 class star = < {WN1#star}, star , "(astronomy) a celestial body of hot gases ...">
binaryStar ... supernova
actor
s2 class star = < {WN5#star,WN7#lead WN3#principal }, star , "an actor who plays a principal role">
co−star ... filmStar
star as property
domain(hotel)
s3 property star = < {TravelOntology#star}, star , "quality of a hotel">

Fig. 2. Possible senses for keyword *star*.

– Keyword Senses Enrichment and Removal of Redundancy: As the obtained senses were built with terms coming from different ontologies, they could represent the same semantics. An incremental algorithm is used to remove possible redundancies, aligning the different keyword senses and merging them when they are similar enough. The senses are merged when the estimated *synonymy probability* between them exceeds a certain threshold. Thus, the result is a set of *different* possible senses for each user keyword entered.
– Disambiguation of Keyword Senses: The system obtains the most probable intended sense of each user keyword considering the possible senses of the rest of keywords. Using a semantic relatedness measure based on information provided by a traditional search engine such as Google or Yahoo!, it computes the correlation between the senses of a particular user keyword and the senses of its neighbour keywords. Thus, the best sense for each keyword will be selected according to its context. Note that this selection can require the user's feedback to select the most appropriate sense for each keyword in a semi-automatic way.

This discovery and disambiguation algorithm, which due to space limitations has been summarized here, is thoroughly described in [21], and has been applied successfully to the integration of senses in semantic repositories [10].

4 Ontology Integration

The result of the previous step is a *semantic keyword* (a keyword and its selected sense) for each keyword input by the user. These semantic keywords have inner information about the *ontological context* of themselves, this is, the semantic information that has been consulted during their construction and that defines them. This information is multiontology-sourced and has been merged during the keyword disambiguation step. In this section, we present the different levels of information that are used to enrich and complete the information about the semantic keywords in the final ontology, and then we give an insight of the method that is used to integrate it.

4.1 Ontological information considered

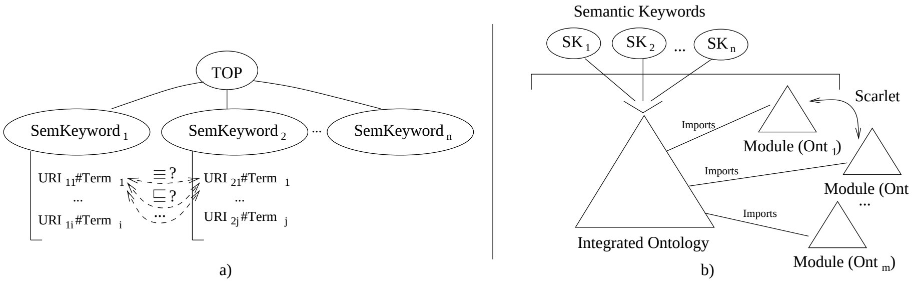


Fig. 3. a) The system has to discover the possible missing relationships. b) This is done by consulting different information sources, which allows our system to enrich and integrate our resulting ontology.

Once the input keywords and their corresponding senses have been disambiguated, our multi-sourced ontology can be integrated. In Figure 3.a, the situation that our system confronts is depicted. The information retrieved to obtain the semantic keywords is stored in their ontological contexts, and, in principle, they are isolated one from another even when they might share sources. Thus, the relationships between ontological contexts can be stated at different levels:

- Sharing a term from an ontology: It is very frequent that, when disambiguating a set of keywords, the same ontology is consulted and different ontological contexts share terms as they affect different keywords definitions.
- Sharing source ontologies: When two ontological contexts do not share ontological terms, they might still share source ontologies. This has to be taken into account as the keywords that are being disambiguated might belong to the same domain.
- Sharing concepts: Even when neither a term nor ontology is shared, there might exist further relationships such as synonymy between the terms of different ontologies.

So, to avoid the possible isolation, and to find out the possible relationships and enrich the information in the final ontology, the system considers three levels of ontological information, as suggested in [3]:

1. *Semantic keyword information.* It is the skeleton of the resulting ontology, and, as it contains the original URIs of the terms involved in each sense

definition, it allows to extract more information from the original source ontologies. This information is asserted altogether in one knowledge base and enriched with the information further retrieved. If no more information should be available, the resulting ontology would have to be considered as light weighted one (shallow ontology). If two ontological contexts share an ontological term, it is stated at this information level.

2. *Automatic modularization and ontology reuse.* To obtain the original definitions of the ontological terms involved in the ontological contexts, the system uses ontology modularization techniques [9]. In brief, given a set of terms of an ontology, the *Extractor* should permit us to obtain a module that is equivalent to the whole ontology regarding what can be inferred about the set of terms given. This allows our system to obtain the complete definitions of the terms and all the existing relationships between terms coming from the same ontology (intra-ontology relationships).
3. *Simple inter-ontology relationships.* Finally, the system can discover inter-ontological relationships using Scarlet [20]. Scarlet is able to obtain disjointness, inheritance and named relations information (i.e. roles that has the pair of concepts as domain and range or vice versa). It also gives an explanation about how the relationship has been found, and our system only includes it if the reasoning path uses ontologies included in the ontological context.

The possible redundancies in the information (information of different levels may overlap) of the resulting ontology are automatically removed with the help of a DL reasoner. By asserting all the information and classifying it, we obtain a final version of the ontology.

4.2 Insight of the method

We now turn our attention to the integration algorithm itself. To achieve a better understanding of the algorithm, the general structure of the integrated ontology is shown in Figure 3.b. The semantic keywords set $\{SK_i\}$ is the result of the keyword disambiguation process, and the input of this step. The algorithm is shown in Algorithm 1. It assumes the existence of a global storage that is used to track the source ontologies and the terms that appear in the ontological contexts of the senses (*ontologyURIStorage*), and a reasoner that is finally used to filter out the possible redundancies.

First, after initializing the storages (lines 2-3), our system translates each of the semantic keywords into OWL (lines 4-6) and stores their translation. During this process, all the URIs of the source ontologies are registered in the *ontologyURIStorage*, and, along with each of them, a list of all the terms that are defined in each ontology is stored. Each of these lists is the signature that the extractor uses to obtain the corresponding ontology module (lines 7-11). In particular, our prototype uses ProSÉ [14], although the method is designed to work with other module extractors. Then, the translations are written in the main module together with the *import* axioms that will include the different modules in the final ontology (lines 12-13).

Algorithm 1 Integration Algorithm

```
1:  procedure sensesToOnt (Array semKeywords, Extractor extractor)
2:  ontologyURIStorage.clear()
3:  translations.clear()
4:  for all sk in semKeywords do
5:      translations.add(translateSense(sk, ontologyURIStorage))
6:  end for
7:  for all uri in ontologyURIStorage.getOntologies() do
8:      // each module is stored separatedly
9:      tmpModule = extractor.getModule(uri, ontologyStorage.getSignature(uri))
10:     writeDownModule(tmpModule)
11: end for
12: writeDown(translations) // we write down all the translations in the main ontology
13: writeDown(importAxioms) // we write the import axioms to include the modules
14: // we finally use Scarlet to obtain inter-ontology relationships
15: scarletInformation = filter(Scarlet.obtainRelationships(ontologyURIStorage));
16: writeDown(scarletInformation)
17: reasoner.classify(newlyIntegreatedOntology)
18: reasoner.writeDown(ontologyWithoutRedundancies)
19: end procedure
```

In this stage, Scarlet is used to discover possible relationships between terms in different ontology modules (lines 15-16). Scarlet can discover relationships following paths between different ontologies, establishing the relationships between terms of two ontologies through relationships with terms of another ones. To avoid introducing ontologies that our system has not processed, it filters out paths to only accept direct relationships (source and target terms belong to ontologies that have been registered in the *ontologyURIStorage*).

Finally, we use a DL reasoner to classify the ontology (lines 17-18) and then write down the classified ontology. This allows us to: 1) eliminate redundant axioms, and 2) detect possible inconsistencies in the integrated ontology and inform the user about them.

5 Application Scenario

To evaluate the feasibility of our approach, we have developed a prototype and applied it to the field of e-commerce. Let us imagine a user that has an e-book store in the Web. This user wants to have the contents on his page annotated semantically to allow the Web robots to index them. However, he is not an expert and does not know any ontology about e-commerce. He introduces the input keywords *book* and *offer* because he wants to start a sales campaign.

Figure 4 shows the intermediate results for these keywords. In our prototype, we have used a controlled set of ontologies to trace and repeat experiments. This set contains the test collection OWLS-TC4[1] plus the ontology *schema.org*[2]. The

[1] `http://projects.semwebcentral.org/projects/owls-tc/`

[2] This ontology is supported in their Web searchers by Google, Yahoo and Microsoft.

ontology obtained for this input can be consulted at `http://sid.cps.unizar.es/ontologies/integration_book_offer.owl`.

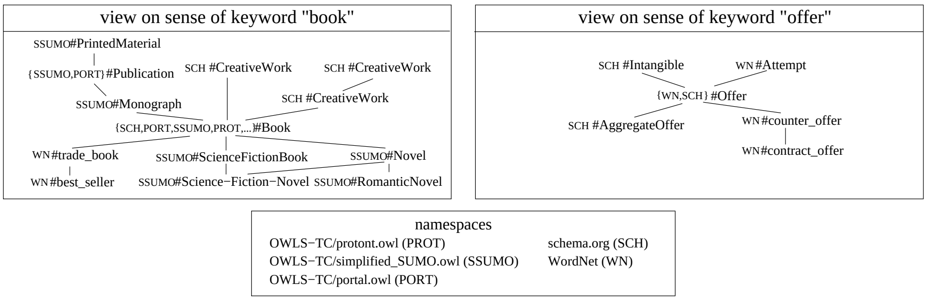


Fig. 4. Excerpts of the senses obtained for “book” and “offer”

Note that both semantic keywords share source ontologies (schema.org, for example), which is exploited by our system to obtain a richer module from them. The user can now use this ontology to add semantic annotations to his site, and, as the integrated ontology keeps the original sources, the robots visiting the site will understand these annotations.

6 Related Work

Different methodologies have been proposed in the last decade for the creation of ontologies [7]. However, these methodologies often require ontological engineers and domain experts, a requirement that our approach gets rid of. Moreover, it is widely recognized that developing an ontology from scratch is a complex and time-consuming task [6]. That is why we adopted the way of reusing ontologies for our approach as a mean to lessen the efforts of ontology construction. In [16], several features that should characterize an integrated environment that helps the construction of ontologies are identified, such as the support of a sophisticated methodology and the possibility of collaborative development. In our approach the collaboration is indirectly achieve by automatically discovering third-parties' ontologies and extracting information from them.

The so-called *ontology learning* [5], which attempts to help in the semi-automatic construction of ontologies, has attracted considerable research interest, as it can reduce significantly the cost of ontology building. In particular, the generation of ontologies from text [4, 6] is a research avenue that has been explored in several works. In our opinion, the main drawback of these approaches is that they build a new ontology that is not directly linked with well-known resources. Besides, they need an important corpus of documents to generate the ontologies. In comparison with these works, the approach presented in this paper reuses existing available ontologies by using semantic techniques to automatically build ontologies from user keywords, integrating information retrieved from well-known sources. So, our approach follows the spirit of the proposal of the position paper [1].

Regarding the first step of our system, a wide range of works have been devoted to word sense disambiguation [17]. In particular, the approach used in this paper for keyword disambiguation has been successfully used in several other tasks, such us ontology matching [11] and sense clustering [10].

7 Conclusions and Future Work

In this paper, we have proposed an approach to ease the development of ontologies and to facilitate their reuse and spread. The user does only have to input the keywords that s/he wants to be included into the ontology and select their meaning among the ones proposed by the system, which disambiguates, merges and retrieves more information about them. This makes the approach to ontological construction envisioned in position paper [1] become real. The proposed system has the following features:

- It consults a dynamic pool of existing ontologies to disambiguate the keywords input by the user.
- Once the user has chosen the exact meaning of each keyword, it integrates the retrieved information into one ontology that can be used as it is or as starting point to refine it.
- It uses different ontological engineering techniques to further improve the information that is integrated into the resulting ontology.
- Indirectly, it facilitates the reuse and spread of existing ontologies, instead of creating new ontologies that should be mapped against the existing ones.

Our main contribution is to use preexisting techniques with a new objective: to ease the building of ontologies from scratch. Using this approach, the initial efforts of developing an ontology are lessened drastically (the user has not even to look for appropriate ontologies, as the system does this task on her/his behalf). As future work, we are planning to study the impact of the quality of the ontologies used as information sources and introduce a method to discriminate the worst ones to refine our resulting ontologies.

Acknowledgments

This work has been supported by the CICYT project TIN2010-21387-C02-02.

References

1. H. Alani. Ontology construction from online ontologies. In *Proc. of 15th Intl. Conf. on World Wide Web (WWW'06), UK*, pages 491–495. ACM, 2006.
2. F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Pastel-Scheneider. *The Description Logic Handbook. Theory, Implementation and Applications.* Cambridge University Press, 2003.
3. C. Bobed and E. Mena. Enhancing the discovery of web services: A keyword-oriented multiontology reconciliation. In *Proc. of 16th Intl. Conf. on Parallel and Distributed Computing (PDPTA'10), USA*, pages 724–730. CSREA Press, 2010.
4. P. Buitelaar, P. Cimiano, and B. Magnini. *Ontology Learning from Text: An Overview.* IOS Press, 2005.

5. P. Cimiano, A. Mädche, S. Staab, and J. Völker. Ontology learning. In S. Staab and D. Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 245–267. Springer, 2009.
6. P. Cimiano, J. Völker, and R. Studer. Ontologies on demand? – a description of the state-of-the-art, applications, challenges and trends for ontology learning from text. *Information, Wissenschaft und Praxis*, 57(6–7):315–320, 2006.
7. O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & Knowledge Engineering*, 46(1):41–64, 2003.
8. M. d'Aquin, C. Baldassarre, L. Gridinoc, S. Angeletou, M. Sabou, and E. Motta. Characterizing knowledge on the Semantic Web with Watson. In *Proc. of 5th Intl. Workshop on Evaluation of Ontologies and Ontology-based tools (EON'07), at ISWC'07, South Korea*. CEUR-WS, 2007.
9. M. dÁquin, M. Sabou, and E. Motta. Modularization: a key for the dynamic selection of relevant knowledge components. In *Proc. of the 1st Intl. Workshop on Modular Ontologies (WoMO'06), at ISWC'06, USA*. CEUR-WS, 2006.
10. J. Gracia, M. d'Aquin, and E. Mena. Large scale integration of senses for the Semantic Web. In *Proc. of 18th Intl. Conf. on World Wide Web (WWW'09)*, pages 611–620. ACM, 2009.
11. J. Gracia and E. Mena. Ontology matching with CIDER: Evaluation report for the OAEI 2008. In *Proc. of 3rd Ontology Matching Workshop (OM'08), at ISWC'08, Germany*. CEUR-WS, 2008.
12. T. R. Gruber. Towards principles for the design of ontologies used for knowledge sharing. In N. Guarino and R. Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation, Deventer, The Netherlands*. Kluwer Academic Publishers, 1993.
13. J. Hendler. Agents and the semantic web. *IEEE Intelligent Systems*, 16(2):30–37, 2001.
14. E. Jimenez-Ruiz, B. Cuenca-Grau, U. Sattler, T. Schneider, and R. Berlanga. Safe and economic re-use of ontologies: A logic-based methodology and tool support. In *Proc. of the 5th European Semantic Web Conference (ESWC'08), Spain*, pages 185–199. Springer, 2008.
15. G. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), November 1995.
16. R. Mizoguchi and K. Kozaki. Ontology engineering environments. In S. Staab and D. Rudi Studer, editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 315–336. Springer, 2009.
17. R. Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41:10:1–10:69, February 2009.
18. H. S. Pinto and J. P. Martins. A methodology for ontology integration. In *Proc. of the 1st Intl. Conf. on Knowledge Capture (K-CAP'01), Canada*, pages 131–138. ACM, 2001.
19. H. S. Pinto, A. G. Pérez, and J. P. Martins. Some issues on ontology integration. In *Proc. of the IJCAI'99 Workshop on Ontologies and Problem Solving Methods, Sweden*. CEUR-WS, 1999.
20. M. Sabou, M. d'Aquin, and E. Motta. SCARLET: SemantiC relAtion discoveRy by harvesting onLine onTologies. In *Proc. of the 5th European Semantic Web Conference (ESWC'08), Spain*, pages 854–858. Springer, 2008.
21. R. Trillo, J. Gracia, M. Espinoza, and E. Mena. Discovering the semantics of user keywords. *Journal on Universal Computer Science*, 13(12):1908–1935, 2007.