Towards Retrieving Scholarly Literature via Ontological Relationships

Vojtěch Svátek, Ondřej Šváb

Department of Information and Knowledge Engineering,
University of Economics, Prague, W. Churchill Sq. 4, 130 67 Praha 3, Czech Republic
{svatek, svabo}@vse.cz

Abstract. We analyse the problem of retrieving scientific literature related to a problem with complex description, and outline the skeleton of a solution. The proposed mixture of methods and approaches covers manual as well as automatic methods, with emphasis on community tagging, automated ontology learning from text and ontology mapping. Symbiosis of RDF/OWL and Topic Maps as underlying formalisms is foreseen. As a very simple proof of concept, relational annotation of five research papers has been carried out independently by two annotators, and the results were analysed.

1 Introduction

Search for specialised (e.g. scholarly) documents, either in digital resources or even in the open web space, differs in many aspects from the 'mainframe' web search by casual users. Experience accumulated in many projects indicates that most web searches for unseen information resources[1] aim at either (or combination of):

- Getting the explanation of an unknown *term*, e.g. a word user's doctor used in his report
- Getting start-up literature for learning about a more general *topic*, e.g. again related to the user's health, or to his/her professional or leisure activities
- Getting an answer for a specific *factual* query (e.g. what is the currency used in the country s/he is travelling to).

While for the first two, *keyword-based* search enhanced with *link popularity* measures typically achieves very good results, the third can be to some degree supported by the construction of large but scope-restricted knowledge bases populated using *wrapper-based information extraction*. 'Semantic' approaches to web search, in this context, thus often merely amount to *disambiguation of individual words*, as in the notorious 'Jaguar' or 'Madonna' examples.

In contrast to more casual term-, topic- or fact-oriented searches, a *researcher* is often interested in resources that deal with certain *problem* and *method* of its solution that is similar to the problem/method s/he is studying him/herself. The search goal is then of more structured nature than that behind term/topic searches, but cannot rely

[1] Such queries being called *informational*, in contrast to navigational or transactional ones [1].

© Václav Snášel (Ed.): Znalosti 2008, pp. 256–267, ISBN 978-80-227-2827-0.
FIIT STU Bratislava, Ústav informatiky a softvérového inžinierstva, 2008.

on existing semi-structured resources. What matters are not just the topics/terms but also their mutual *relationships* with respect to e.g. procedures, experiments or observed events described in the resources.

Imagine a PhD student in Computer Science who would like to investigate the possibility of using an information extraction tool based on statistical methods in order to acquire background knowledge from Wikipedia, which will in turn be used within an adaptive e-learning system. Plain conjunctive queries for terms such as "statistical", " information extraction", "background knowledge", "Wikipedia", "e-learning" allow for numerous interpretations, such as "Wikipedia pages about information extraction", "e-learning course about statistical methods" or "information extraction using background knowledge". In the web context, moreover, it is quite likely that link-popularity measures will actually give higher rank to well-known combinations of topics rather than to those on the cutting edge of research.

A tentative example of *relation-centric* representation of the above query could look as follows:

```
TOOL1 has_type tool/method
TOOL1 based_on_formalism Statistics
TOOL1 applies what:Information_extraction  on:Wikipedia
TOOL1 produces  what:RESOURCE1  from:Wikipedia
RESOURCE1 has_type  resource/data
TOOL2 has_type  tool/method
TOOL2 has_feature  Adaptivity
TOOL2 uses  what:RESOURCE1  purpose:E-learning
      role:Background_knowledge
```

On example we illustrate a few features our relational language should possibly satisfy:

1. It should allow to assign at least a few *types* ('tool/method', 'resource/data') to entities the paper would essentially be about, i.e. some tool/method (TOOL1, TOOL2) or resource (RESOURCE2) denoted by variables. Note that in the *annotations* these could either correspond to constants (names of tools, resources etd.), or could remain 'anonymous'if the entities don't have specific names.
2. It should allow to express *n-ary relationships* (e.g. that a tool uses a resource in some role and/or for some purpose). N-ary relationships (with n>2) are here 'applies' and 'uses'.
3. It should allow to express the *roles* of the entities in such a relationship, in order to distinguish that something ('what') is applied on something ('on')[2].

While the first feature is inherent to semantic web technologies, the remaining two are not directly present in semantic web languages; we return to this issue below. The example also contains numerous redundancies, which could also arise in reality, as result of navigation-based query formulation, also see below.

Note that adding names of relations to the original *keyword-based* query alone would not help much, as these are mostly quite common words. On the other hand,

[2] Note that one of the roles in another relationship ('uses') is accidentally named 'role', too.

in a *relation-centric* representation, the problem of parallel vocabularies used by different communitites (e.g. machine learning vs. statistical prediction) could be partially alleviated, as the 'relational' notions of e.g. 'producing something as result' or 'using something as instrument' are quite likely to be labelled consistently.

Although relation-centric representation of content is more powerful than keyword-based one, someone may argue that even more sophisticated representation would be desirable, e.g. based on a pre-defined ontology of hypotheses, procedures, steps, experiments, claims etc. (possibly using respective upper-level ontologies as starting point [6]). However, we are afraid that *insisting on* populating such a complex model with resource annotations would go too long a way from the common practice of most non-AI-oriented users, who are only used to keyword/topic labelling in digital libraries, folksonomies etc., and would be unlikely to be adopted in larger scale. On the other hand, capturing information among entities or topics using (binary or n-ary) relationships is supported by built-in features of many formalisms, such as ER models, object models, RDF, Topic Maps and the like, and should thus be familiar to all medium-degree computer-literate users. In addition, a complex model could appear not well portable to different domains. Moreover, considering such models could be even unnecessarily restrictive in terms of recall, as relevant related work to the user's research will quite often be any work dealing with the same or similar problem, even if it proceeds in a different way or rises different claims about the problem. And, finally, a simple representation could be extended in the future in a bottom-up fashion, in the sense of promoting the most frequently used relationships to stable elements of the language.

The long-term goals of the research direction outlined in this paper are:

- to investigate the feasibility and efficiency of relation-centric annotation of scholarly resources
- to design a collection of tools supporting easy authoring of annotations
- to design a search tool leveraging on annotations
- to test all above in several real domains.

The remaining text is structured as follows. Section 2 lists some general problems encountered by prior research, and our proposed solutions. Section 3 reports on an initial experiment with relational annotation. Finally, section 4 compares our approach with some other projects, and section 5 wraps up the paper.

2 Obvious Problems and Proposed Solutions

In this section we try to enumerate problems related to semantic annotation of resources in general, and attempts to formulate remedies to them.

Problem: People generally find manual semantic annotation tedious.
Solutions:

- Semantic annotation embedded in the document itself can be leveraged, which may have been provided by the author him/herself using approaches such as SALT [6].

- The annotation can arise as by-product of paper submission to an event (enforced to the author by the submission form) or of paper review (asked from the reviewer by the review form)
- Concepts and relations will be suggested via NLP-based analysis (ontology learning / information extraction, see e.g. [2, 8]) of the document content, with specific focus on the logical structure (esp. headings), relying on discourse analysis principles.
- For given concepts, relations will be suggested based on the relations used for these or more general/specific concepts by other members of the community, cf. [12].
- Concepts and relations can partly be ported from earlier work referenced by the paper, in particular, in the 'Related Research' section. NLP can also be used here so as to reflect fine-grained distinctions, especially, the *difference* from earlier work explicitly stated by the authors. It is obvious that the degree of 'machine understanding' will be limited, as capturing the crucial relationships in a research paper are often difficult even for a human. However, note that we are not aiming at populating a knowledge base for the purpose of automated reasoning but merely at providing more accurate search than plain keyword-based one; it is likely that even search results with precision below 50% could supersede such a baseline.

Problem: People find semantic web technology complicated or the expressive power of semantic web languages inadequate.
Solution: In addition to RDF/OWL technology (which has been by now endorsed, to some degree, by the computer science community), use alternatively the Topic Maps[3] technology for user communities who would prefer it (e.g. in library-oriented computing or beyond the computing realms as such). This will bring the following benefits:

- Topic Maps have simpler syntax, as they are expressed in native XML. Their code can be easily read by humans.
- Topic Maps allow to directly express n-ary relations. In OWL this can be done using design patterns [11], however, this typically leads to loss of modelling clarity, as reified relations form 'unnatural' concepts.
- Topic Maps allow to directly express associations with roles. The use of roles is natural for many users in conceptual modelling, as it is compatible with e.g. UML class diagrams. In OWL this can be done e.g. using the role pattern [14], but full realisation of roles requires the use of the SWRL rule language on the top of OWL.

The transformation between semantic web languages and Topic Maps has to be assured, which should be transparent for the end user. There has already been some work on comparing RDF/OWL technology and making them interoperable [5, 13] but few projects where such interoperability really materialised. In this context, the proposed work would have pioneering aspects in general, and would be highly valuable for promoting further cooperation between the two sibling communities, which have both important positions in various application domains (also thanks to being endorsed by the W3C and ISO, respectively).

[3] www.topicmaps.org

Problem: Community tagging of any kind leads to heterogeneity of entity labels. Note that even in our simple example, there have been used two relations with very similar semantics: 'applies' and 'uses'.
Solutions:

- State-of-the-art methods of ontology mapping [3] and reasoning by analogy [9] techniques can be applied so as to cluster entities with the same meaning.
- In a more top-down fashion, well-elaborated ontology content design patterns [4] could be used as common denominator of the structures of concepts and relations to be mapped.
- The principles of *navigation-based* search query formulation should enable the user select the right relations. The ontology would be displayed and the user could explore it, while marking a subgraph of the whole ontology - concepts and relations.

It is important to state that RDF/OWL and Topic Maps rather represent technological platforms. The design of higher-level semantic structure of the relational representation will require to study numerous prior approaches both in the field of knowledge modelling and (semantic) language modelling, the latter for the sake of making easier the mapping from relational structures returned by NLP methods to user-oriented general relationships.

3 Initial Experiment

In this section we describe an experiment that was meant as an initial proof of concept, at least for a tiny part of the whole intended architecture—namely, for the expressivity of the simple representation chosen with respect to real scholarly articles. The experiment was carried out using the following steps:

1. A seed vocabulary was set up. For simplicity, we took the example of query presented in section 1.
2. We put together a handful of simple, tentative guidelines for relational annotation.
3. Each of the authors (later referenced as VS and OŠ, respectively), entirely separately, annotated the same five papers from a major conference. The annotation was based on reading the papers, and had to reflect the guidelines and the example query.
4. We computed a simple statistics of the annotations, by each annotator separately, with special focus on the usage of relations from the seed vocabulary vs. introduction of new ones.
5. We also analysed the evolution of the graph structure of the underlying relational vocabulary, attempted to informally match the two diverging (annotator-specific) versions of the vocabulary, and highlighted the points interesting with respect to possible merging.

Each step will be described in a separate subsection.

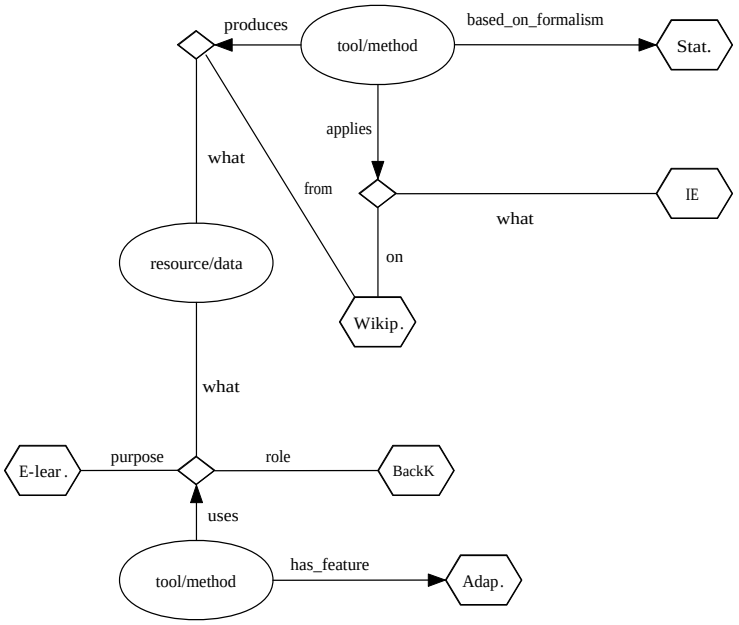

Fig. 1. Graphical representation of the example

3.1 Seed Vocabulary

Rather than attempting to systematically design a vocabulary to be used, we decided, for simplicity, to reuse the example that was originally meant as a motivating one. This is however consistent with the spirit of the whole envisaged project: we assume that wide-scale annotation should rely on ad hoc annotators who might not wish to spend their time getting acquainted (and being constrained) with elaborate ontologies. The fact that the example was conceived as 'query' did not harm either, as we assume the language of annotations and queries to be principially the same (possibly with the exception of concrete names of named entities, see below). The example is depicted graphically in Fig. 1; the graphical language was currently designed ad hoc, but hopefully borrows from common graphic languages enough for being well-readable. Unspecified entities, corresponding to variables in the pseudo-language, are depicted as ovals labelled with the type of the entity. Specified entities are depicted as hexagons, labelled with the entity name. Relations are depicted as directed arcs labelled with the relation name; furthermore, for relations with arity higher than 2, the labelled arc ends in a diamond, from which undirected edges corresponding to roles lead to the respective arguments of the relation.

The seed vocabulary thus consisted of five relations—'based_on_formalism', 'applies', 'produces', 'has_feature' and 'uses', plus the (unique) construction 'has_type' assigning a type to a variable, and two general types of entities: 'tool/method' and 'resource/data'. It is depicted in Fig. 2. Compared to the example from which it was

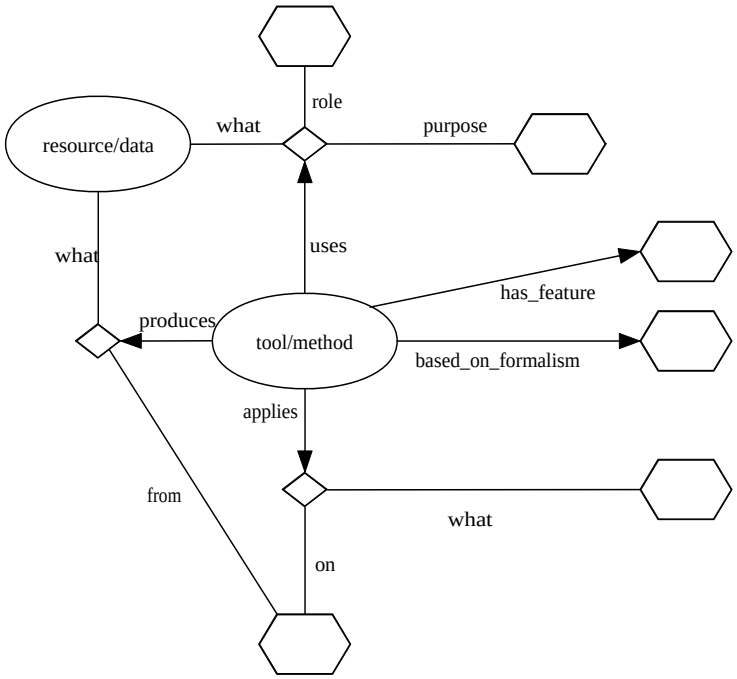

Fig. 2. Graphical representation of the seed vocabulary

abstracted, the 'type' nodes are merged, and the names of specific entities (instances) are stripped off.

3.2 Annotation Guidelines

To provide for minimal degree of shared understanding of the annotation task, we formulated brief guidelines. They consisted in the following:

- The number of relation instances per annotated paper should be around 10-20.
- Relations from the seed example are to be reused if considered fully adequate. Otherwise new relations can be introduced without any reservation.
- The annotations should be relation-centric where possible. Entities eligible for relations should be modelled as such, rather then being 'reified' to concepts. N-ary relations should be used where appropriate.

The guidelines were not binding, and indeed the first (probably, least important) one was not literally obeyed, as one of the annotators (VS) eventually used fewer relation instances than the recommended number, in most cases.

3.3 Actual Annotation

As material for annotation we took five[4] papers from the recent 4th European Semantic Web Conference, which we actually attended. This setting implied that the annotators (us) were to some degree acquainted with the topics addressed by the papers, but were not their authors. The time needed to accomplish one annotation (briefly read the paper and create the relational representation) floated between 20–30 minutes, but, did not seem to be affected by whether the annotator actually attended the respective talk at the conference or not (clearly because there has been a several months' delay).

It should be noted that the relatively high cost of obtaining the annotations observed in the experiment is not a fixed feature of the proposed approach. The proposed settings of manual annotation (by authors, by reviewers and by casual readers) assume the annotator to be familiar with the paper content for a different purpose than the annotation itself.

3.4 Annotation Statistics

The summary information is contained in Table 1. It lists the (maximal, minimal and average) numbers of relation instances, computed for each annotator along the all five annotations, and the numbers of newly-introduced relations and types.

Table 1. Annotation statistics

Annotator	VS	OŠ
Max. no. of relation instances	13	22
Min. no. of relation instances	5	8
Avg. no. of relation instances	7.2	13.0
New relations introduced	6	9
New types introduced	0	4

Furthermore, Fig. 3 depicts the evolution of the number of vocabulary constructs, i.e. (the sum of) relations and types. The plots are given for each of the two annotators plus the union of both[5]. We can see that, expectedly, the number of newly introduced constructs decreases after the first couple of annotations. It would however be interesting to see how strongly this will be affected by crossing various types of domain boundaries.

3.5 Evolution of Graph Structure

Fig. 4 shows examples of vocabulary extensions accomplished by VS in graphical form, marked in bold. These consist in new *relations* 'applied_on' and 'analyses', and a new *role* 'to' in the existing relation 'applies'. The extensions accomplished by OŠ are

[4] The first paper from each of the first five sections of the proceedings was chosen.

[5] For the newly introduced constructs, the intersection only contained one relation: 'has_name', which should actually not be treated as relation proper in the future version of the language.

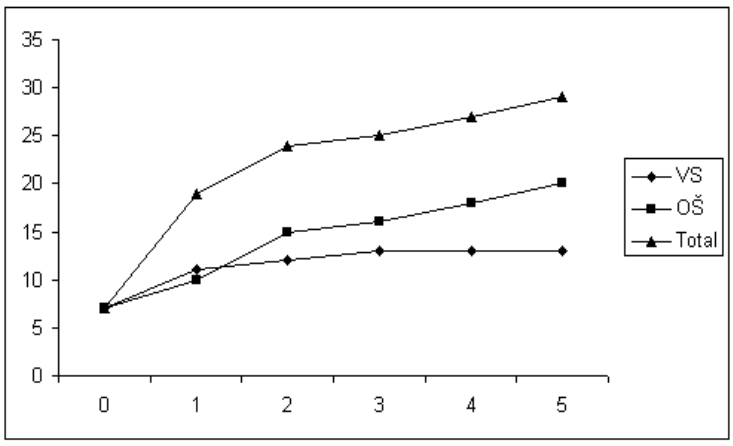

Fig. 3. Evolution of number of relations+types

harder to display using a quasi-planar graph, as this annotator decided to introduce new types (such as 'project' or 'methodology'), which made the network more intermingled. We omit it due to space reasons.

There were numerous cues on how relations, roles and types could be merged in an integrated model. For the sake of brevity, we will only discuss one example. For one of the papers, the annotation by VS contained, among other, the statements

```
METHOD1 has_type tool/method
METHOD1 produces what:Ontology from:Source_code
```

and the annotation by OŠ contained, among other, the statements

```
TOOL2 has_type tool
TOOL2 applies what:code_analysis on:source_code
```

We can see that if we consider the type 'tool/method' as overlapping with 'tool', then it makes sense to derive that in order to 'produce' something from some entity, we may 'apply' something on it. A bit more formally, we can hypothesise about the dependency between the following two patterns:

$$X \in \text{`}tool/method\text{'} \wedge X \in \text{`}tool\text{'}$$

$$(X\ applies\ what : Y\ on : Z) \wedge (X\ produces\ what : W\ from : Z)$$

Namely, if one of the patterns holds then the probability of the other to hold is increased as well.

4 Related Work

Our suggested approach informally builds on experiences reported from several known literature annotation projects. However, the mix of complementary and supplementary techniques seems to be unique.

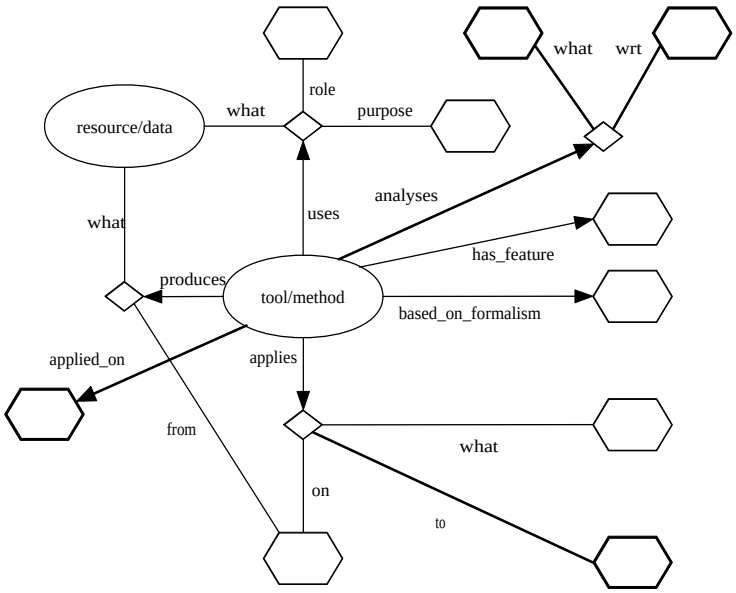

Fig. 4. Examples of vocabulary extension

Approaches purely based on *manual annotation*, such as [6], can be much more 'semantic' in terms of capturing sophisticated details of the argumentation in the annotated document. However, their scalability is an issue. Furthermore, as mentioned before, detailed argumentation and derived conclusions are not indispensable parts of annotation in our setting. It is quite possible that the user would like to retrieve as many papers dealing with the particular structure of problems, tasks, methods, experiments, tools etc., whatever the motivations or final outcomes of the research are.

In contrast, approaches focussing on automated analysis via *NLP techniques* retain the bottleneck of domain-neutral text analysis: the annotation has low precision and is tighly bound to the vocabulary used by the given community. Projects such as [10] indeed aim at the development of community-specific portals rather than at cross-domain retrieval of relevant literature.

The best-known present *collaborative* literature annotation tool, Bibsonomy [7], relies on a conventional system of tags known from common folksonomies. The possibility of relational tagging has not been considered yet.

Finally, a practical innovation of our proposal lays in the symbiosis of *RDF/OWL* and *Topic Maps* technologies, with the possibility of combining their comparative advantages. Although the interoperability of the two has sometimes been declared, there are to date very few projects in which it truly materialised. However, our conviction is that the disconnection is essentially due to social and historical reasons, and could be overcome with reasonable effort.

5 Conclusions and Future Work

We presented a possible way to improve the retrieval of scholarly literature across domain boundaries. The most important ingredients of the approach are the light-weighted relational representation and the use of a diverse range of annotation methods, including authoring-stage annotation, reviewing-stage annotation, end-reader-based manual annotation leveraging on the social bookmarking paradigm, automated (NLP-based) annotation, and usage of general ontology mapping and ontology design patterns techniques. A tiny initial experiment has been accomplished, which merely covered the knowledge representation and end-reader-based manual annotation aspects of the whole approach.

Future work should, simply spoken, extend the approach along all of its axes. Based on the lessons from the initial experiments, the representation language will achieve firmer and formal (though not necessarily final) shape, and the annotation guidelines will be amended (but without significantly increasing their size). An important step is to identify the way to introduce the lightweight relational representation into existing social bookmarking frameworks. This would allow for larger-scale and more objective experiments, with annotators unbiased by knowledge engineering background, and possibly even addressing documents from outside computer science. Finally, the most interesting and challenging issues from the technical and theoretical point of view are related to automated methods of annotation and vocabulary graph matching.

6 Acknowledgments

The research leading to this paper was partially supported by the IGA VSE grant no.12/06 “Integration of approaches to ontological engineering: design patterns, mapping and mining”, by the IGA VSE grant no.20/07 “Combination and comparison of ontology mapping methods and systems”, and by the Knowledge Web Network of Excellence (IST FP6-507482). The authors would like to thank Jirka Kosek and Pavel Smrž for discussion over the initial ideas behind this paper.

References

1. Broder, A.: A taxonomy of web search. *ACM SIGIR Forum archive*, Volume 36 , Issue 2, 2002.
2. Cimiano, P.: Ontology Learning and Population from Text: Algorithms, Evaluation and Applications. Springer, 2006.
3. Euzenat, J., Shvaiko, P.: Ontology Matching. Springer, 2007.
4. Gangemi, A.: Ontology Design Patterns for Semantic Web Content. In: The Semantic Web - ISWC 2005, 4th International Semantic Web Conference, Springer LNCS 3729.
5. Garshol, L. M.: Living with topic maps and RDF. Online `http://www.ontopia.net/topicmaps/materials/tmrdf.html`.
6. Groza, T., Handschuh, S., Möller, K. H., Decker, S.: SALT - Semantically Annotated LaTeX for scientific publications. In: Proc. European Semantic Web Conference (ESWC'07), Springer, LNCS 4519, 2007.

7. Jäschke, R., Hotho, A., Schmitz, A., Stumme, G.: Analysis of the Publication Sharing Behaviour in BibSonomy. In: Proc. ICCS 2007, Springer LNCS 4604, 283-295.
8. Kavalec, M., Svátek, V.: A Study on Automated Relation Labelling in Ontology Learning. In: P.Buitelaar, P. Cimiano, B. Magnini (eds.), Ontology Learning and Population, IOS Press, 2005, 44-58.
9. Nováček, V.: Inferential Ontology Learning. In: Knowledge Web PhD Symposium 2007, co-located with the 4th Annual European Semantic Web Conference, Innsbruck 2007.
10. Nováček, V., Smrž, P.: Ontology Acquisition for Automatic Building of Scientific Portals. In: SOFSEM'06, Springer LNCS 3831, 493–500.
11. Noy, N., Rector, A.: Defining N-ary Relations on the Semantic Web. W3C Working Group Note 12 April 2006. Online http://www.w3.org/TR/swbp-n-aryRelations/.
12. Oren, E., Gerke, S., Decker, S.: Simple Algorithms for Predicate Suggestions using Similarity and Co-Occurrence. In: Proc. European Semantic Web Conference (ESWC'07), Springer, LNCS 4519, 2007.
13. Pepper, S., Vitali, F., Garshol, L. M., Gessa, N., Presutti, V.: A Survey of RDF/Topic Maps Interoperability Proposals. W3C Working Draft 29 March 2005. Online http://www.w3.org/TR/2005/WD-rdftm-survey-20050329/.
14. Sunagawa, E., Kozaki, K., Kitamura, Y., Mizoguchi, R.: Role organization model in Hozo. In: Managing Knowledge in a World of Networks (Proc. of EKAW 2006 - 15th International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks), Lecture Notes in Computer Science (LNCS), Springer Verlag, Vol. 4248, pp.67-81, Podebrady, Czech Republic, 2-6 Oct., 2006.