MELIS
An Incremental Method For The Lexical Annotation Of Domain Ontologies

Sonia Bergamaschi, Laura Po, Maurizio Vincini
DII - Università di Modena e Reggio Emilia, via Vignolese 905, Modena, Italy
bergamaschi.sonia, po.laura, vincini.maurizio@unimore.it

Paolo Bouquet, Daniel Giacomuzzi
DIT - Università di Trento, Via Sommarive, 14, Trento, Italy
bouquet@dit.unitn.it

Francesco Guerra
DEA - Università di Modena e Reggio Emilia, v.le Berengario 51, Modena, Italy
guerra.francesco@unimore.it

Keywords: The paper should have at least one keyword. The text should be set in 9-point font size and without the use of bold or italic font style.

Abstract: In this paper, we present MELIS (**M**eaning **E**licitation and **L**exical **I**ntegration **S**ystem), a method and a software tool for enabling an incremental process of automatic annotation of local schemas (e.g. relational database schemas, directory trees) with lexical information. The distinguishing and original feature of MELIS is its incrementality: the higher the number of schemas which are processed, the more background/domain knowledge is cumulated in the system (a portion of domain ontology is learned at every step), the better the performance of the systems on annotating new schemas.
MELIS has been tested as component of MOMIS-Ontology Builder, a framework able to create a domain ontology representing a set of selected data sources, described with a standard W3C language wherein concepts and attributes are annotated according to the lexical reference database.
We describe the MELIS component within the MOMIS-Ontology Builder framework and provide some experimental results of MELIS as a standalone tool and as a component integrated in MOMIS.

1 INTRODUCTION

The growth of information available on the Internet has required the development of new methods and tools to automatically recognize, process and manage information available in web sites or web-based applications. One of the most promising ideas of the Semantic Web is that the use of standard formats and shared vocabularies and ontologies will provide a well-defined basis for automated data integration and reuse. However, practical experience in developing semantic-enabled web applications and information systems shows that this idea is not so easy to implement. In particular, we stress two critical issues: on the one hand, building an ontology for a domain is a very time consuming task, which requires skills and competencies which are not always available in organizations; and, on the other hand, there seems to be an irreducible level of semantic heterogeneity, which has to do with the fact that different people/organizations tend to use "local" schemas for structuring their data, and ontologies – if available at all – are often designed to fit locally available data rather than aiming at being general specifications of domain knowledge. The consequence is a situation where data are organized to comply to some local schema (e.g. a relational schema, or a directory tree) and no explicit (formal) ontology is available; or – if an ontology is available – it is tailored on local data/schemas and therefore of little use for data integration.

The two issues above led the Semantic Web and Database communities to address two very hard problems: *ontology learning* (inducing ontologies from data/schemas) and *ontology matching/integration* (bridging different ontologies). For our argument, we only need to observe that several methods and tools developed to address the two problems rely – in different ways – on the use of lexical information. The reason is simple: beyond the syntactic and semantic heterogeneity of schemas and ontologies, it is a fact that their elements and properties are named using natural language expressions, and that this is done precisely because they bring in useful (but often implicit) information on the intended meaning and use of the schema/ontology under construction. Therefore, it should not come as a surprise

that a large number of tools for ontology learning and schema/ontology matching include some lexical resources (mainly WordNet[1]) as a component, and use it in some intermediate step to annotate schema elements and ontology classes/properties with lexical knowledge. To sum up, lexical annotation seems to be a critical task to develop smart methods for ontology learning and matching.

In this context, we developed MELIS (**M**eaning **E**licitation and **L**exical **I**ntegration **S**ystem), a method and a software tool for the annotation of data sources. The distinguishing feature and the novelty of MELIS is its incremental annotation method: the more sources (including a number of different schemas) are processed, the more background/domain knowledge is cumulated in the system, the better the performance of the systems on new sources. MELIS supports three important tasks: (1) the source annotation process, i.e. the operation of associating an element of a lexical reference database (WordNet in our implementation, but the method is independent from this choice) to all source elements, (2) the customization of the lexical reference with the introduction of new lexical knowledge (glossa, lemma and lexical relationships), and (3) the extraction of lexical/semantic relationships across elements of different data sources.

Works related to the issues discussed in this paper are in the area of languages and tools for annotations ((Bechhofer et al., 2002), (Staab et al., 2001) and (Handschuh et al., 2003) where an approach similar to our is adopted), techniques for extending WordNet ((Gangemi et al., 2003), (Montoyo et al., 2001) and (Pazienza and Stellato, 2006) where a system coupled with Protègè[2] for enriching and annotating sources is proposed), and systems for ontology management (see the the Ontoweb[3] and the Knowledgeweb Network of Excellence[4] technical reports for complete surveys).

2 MELIS: the lexical knowledge component

In most real world applications, ontology elements are labeled by natural language expressions. In our opinion, the crucial reason for this aspect of ontology engineering is the following: while conceptual annotations provide a specification of how some terminology is used to describe some domain (the standard role of OWL ontologies), natural language labels (lexical annotations) provide a natural and rich connection between formal objects (e.g. OWL classes and properties) and their *intended* meaning. The intuition is that grasping the intended interpretation of an ontology requires not only an understanding of the formal properties of the conceptual schema, but also knowledge about the meaning of labels used for the ontology elements. In other words, an OWL ontology can be viewed as a collection of *formal* constraints between terms, whose intended meaning also depends on lexical knowledge.

In most cases, lexical knowledge is used for annotating schema/ontology labels with lexical information, typically WordNet senses. However, lexical annotation is a difficult task, and making it accurate may require a heavy user involvement. Typical problems are: *coverage* (a complete lexical database including all possible terms does not exist); *polysemy* (in natural language, many terms have multiple meanings); *compound terms* (schemas and ontologies are often labeled with compound nominal expressions, like “full professor”, “table leg”, “football team”, and the choice of the right lexical meaning often depends on determining the relationship between terms); *integration* (a standard model/language for describing lexical databases does not exist).

That is why several tools which were developed for annotating sources only provide a GUI for supporting the user in the manual execution of the task. However, this manual work can be highly time consuming, and very tedious for humans.

MELIS tries to make annotation as automatic as possible by providing a candidate lexical annotation of the sources as the combination of lexical knowledge (from WordNet) and domain knowledge (if available). In addition, MELIS uses the *WNEditor* (Benassi et al., 2004) to support customized extensions of WordNet with missing words and senses.

In the following we describe the MELIS method, its heuristic rules and the main features of *WNEditor*.

2.1 The MELIS method

The way MELIS works is depicted in Figure 1. We start from a collection of data sources which cover related domains, e.g. hotels and restaurants. In general we do not assume that a domain ontology is initially available, though this may be the case. The process is a cycle which goes as follows:

1. a schema, which can be already partially annotated with lexical information, is given as input to MELIS, together with a (possibly empty) domain

[1] See `http://wordnet.princeton.edu` for more information on WordNet.

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

[3] http://www.ontoweb.org, in particular deliverable 1.4

[4] http://knowledgeweb.semanticweb.org/

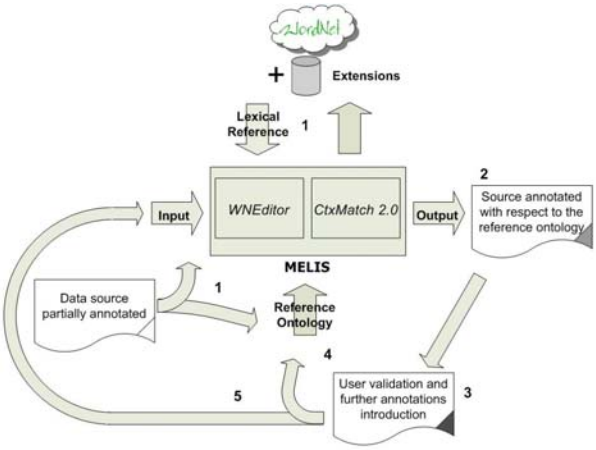

Figure 1: Functional representation of MELIS

ontology (called reference ontology in the figure). Lexical information is extracted from WordNet which may be extended with words/senses which are not available by interacting with *WNEditor*;

2. the automatic lexical annotation process starts; its output is a partial annotation of schema elements, together with a list of discovered relationships across different elements. This annotation, whose main rules are described below, is obtained by using two main knowledge sources: WordNet (for lexical senses and relationships across them), and the reference ontology, if not empty (it provides non-lexical – domain dependent – relationships across senses, e.g. between "hotel" and "price"). Pre-existing lexical annotations are not modified, as they may come either from manual annotation or from a previous annotation round;

3. the resulting annotated schema is passed to a user, who may validate/complete the annotation produced by MELIS;

4. the relationships discovered across terms of the schema are added to the reference ontology (which means that an extended – and lexically annotated – version of the domain ontology is produced, even if initially it was empty);

5. the process restarts with the following schema, if any; otherwise it stops.

The process is incremental, as at any round the lexical database and the reference ontology may be extended and refined. As we said, the process might even start with an empty reference ontology, and the ontology is then constructed incrementally from scratch.

2.2 The rules for generating new annotations

A crucial part of the process has to do with the rules which are used to produce the MELIS lexical annotations. The core rules are derived from CTXMATCH2.0, and are described in previous publications. However, to improve the precision and recall of MELIS, we added a few specialized heuristic rules.

In what follows, we use the following notational conventions:

- Letters: capital letters (A, B, C, ...) stand for class labels, low case letters (a, b, c, ...) stand for datatype property labels, letters followed by "#n" (where n is a natural number) refer to the n-th sense of the label for which the letter stands (e.g. b#2 is the second sense of the word occurring in the label "b").
- Arrows: red arrows denote a subclass relation, black arrows denote datatype properties, blue arrows denote object properties.
- Ontologies: O is used for the ontology to be annotated, DO_i for the i-th domain ontology available for the current elicitation process.

The annotation process takes as input a schema *O* and works in two main steps: first, for every label in *O*, the method extracts from WordNet all possible senses for the words composing the label; then, it filters out unlikely senses through some heuristic rules. The remaining senses are added as lexical annotation. Ideally, the system produces just one annotation, but in general it may be impossible to select a single annotation. Below is a general description of the heuristic rules used by MELIS.

Rule 1 ***Couple class–property****: if in O we find a class labeled A with a datatype property b, and in some DO_i we find a class annotated as A#i and a datatype property annotated as b#j, then we conclude that the annotations A#i and b#j are acceptable candidate annotations for A and b in O.*

Rule 2 ***Inheritance parent–child****: if in O we find a class labeled* `A` *with a datatype property* `b`*, and in some DO_i we find a class annotated as* `B#j` *, with a datatype property annotated as* `b#k` *and a subclass*[5]

[5]In the paper we consider a subclass property both an object oriented definition and a WordNet *hyponym* relation. In WordNet we say that a noun X is a *hyponym* of a noun Y if X is less general than Y (X is a specialization of Y); conversely, we say that X is a *hypernym* of Y if X is more general than Y. Other relationships across nouns are: X is a *holonym* of Y (Y is a part of X), and X is a *meronym* of Y (X is a part of Y). Different relationships are used for

A#i, *then we conclude that the annotations* A#i *and* b#k *are acceptable candidate annotations for* A *and* b *in O.*

Rule 3 ***Inheritance child–parent****: if in O we find a class labeled A with a datatype property b, and in some DO_i we find a class annotated as A#i , with a subclass B#j, and the latter has associated a datatype property annotated as b#k and i, then we conclude that the annotations A#i and b#k are acceptable candidate annotations for A and b in O.*

Rule 4 ***Inheritance in sibling classes****: if in O we find a class labeled A with a datatype property b, and in some DO_i we find a class annotated as C#k with two subclasses annotated as A#i and B#j, and there is a datatype property annotated b#h associated to B#j, then we conclude that the annotations A#i and b#k are acceptable candidate annotations for A and b in O.*

Rule 5 ***Propagation through object properties****: if in O we find a pair of classes labeled A and B, connected through any object property, and in some DO_i we find a pair of classes annotated as A#i and C#k, and C#k has a subclass B#j, then we conclude that the annotations A#i and B#j are acceptable candidate annotations for A and B in O.*

Rule 6 ***Inheritance of parent–child relationship****: if in O we find a pair of classes labeled* A *and* B *(with* B *subclass of* A*), and in some DO_i we find a subclass hierarchy in which two classes are annotated as* A#i, ..., B#j *(with none, one or more intermediate classes in between), then we conclude that the annotations* A#i *and* B#j *are acceptable candidate annotations for* A *and* B *in O.*

When all heuristic rules are applied, then we discharge any candidate pair of annotations which is not supported by any of the rules above.

2.3 The *WNEditor*

WNEditor aids the designer to extend WordNet to cover domain specific words which are not in the original WordNet database.

Since WordNet is distributed *as-it-is* and external applications are not allowed to directly modify its data files, *WNEditor* addresses two important issues: (i) providing a physical structure where WordNet and all its possible extensions are stored and efficiently retrieved; (ii) developing a general technique which can support users in consistently extending WordNet. The first issue is technically solved by storing the original WordNet (and all its possible extensions) in a relational database. The second issue is addressed by giving ontology designers the possibility to *insert new synsets* (a synset is a new "concept", which can be expressed by several words); *insert new lemmas* (based on an approximate string matching algorithm to perform the similarity search on the whole synset network); *inserting new relationships* between synsets (given a source synset, the designer is assisted in searching for the most appropriate target synset, see (Benassi et al., 2004) for details).

WordNet extensions may then be exported and reused in other applications (though they are always presented separately from the original WordNet, and the source of the new annotations is always made explicit).

3 MELIS as a module of MOMIS

We tested the MELIS approach by coupling it with MOMIS[6]. MOMIS is a framework that starts from a collection of data sources and provides a collection of tools for:

1. semi-automatically building a customized ontology which represents the information sources;
2. annotating each source according to the resulting ontology;
3. mapping the created ontology and the original sources into a lexical database (WordNet) to support interoperability with other applications.

MELIS has been experimented in MOMIS to show that it can improve the MOMIS methodology in two main directions: by supporting the semi-automatic annotation of the original data sources (currently the process is manually executed), and by providing methods for extracting rich relationships across terms by exploiting lexical and domain knowledge.

MOMIS provides a double level of annotation for data sources and the resulting ontology: for each source, conceptual annotations map the original structure into a formalized ontology and lexical annotations assign a reference to a WordNet element for each source term. Moreover, the ontology structure is formalized by means of a standard model and each concept is annotated according to a lexical reference. MELIS inside MOMIS allows a greater automation in the process of source annotation, and provides a

other grammatical types, e.g. for verbs and adjectives. See http://wordnet.princeton.edu for more details.

[6] See http://www.dbgroup.unimore.it for references about the MOMIS project.

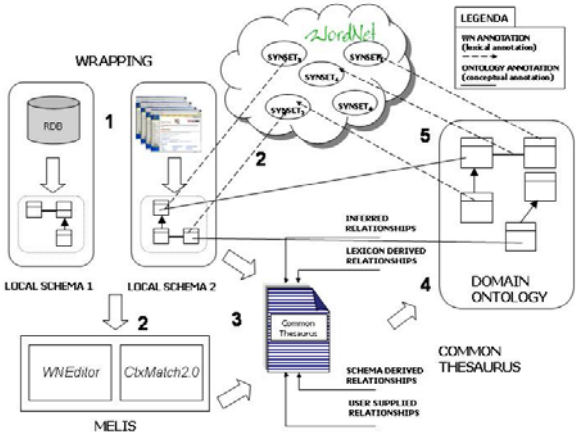

Figure 2: Functional representation of MOMIS and MELIS

way for discovering relationships among sources elements.

Figure 2 shows how the MELIS component is integrated into the MOMIS architecture. The process of creating the ontology and defining the mappings is organized in five step (each task number is correspondingly represented in figure 2) : (1) local source schema extraction, (2) lexical knowledge extraction performed with MELIS, (3) common thesaurus generation, (4) GVV generation, and (5) GVV and local sources annotation. The following sections describe the details of these steps.

Local source schema extraction To enable MOMIS to manage web pages and data sources, we need specialized software (wrappers) for the construction of a semantically rich representations of the information sources by means of a common data model.

Lexical knowledge extraction The extraction of lexical knowledge from data sources is typically based on an annotation process aiming at associating to each source element an effective WordNet meaning.

MELIS supports the user in this task by providing an effective tool for decreasing the boring manual annotation activity.

Common thesaurus generation The common thesaurus is a set of relationships describing inter- and intra-schema knowledge about the source schemas.

The common thesaurus is constructed through a process that incrementally adds four types of relationships: schema-derived relationships, lexicon derived relationships, designer-supplied relationships and inferred relationships.

- **Schema-derived relationships.** The system automatically extracts these relationships by analyzing each schema separately and applying a heuristic defined for the specific kind of source managed.
- **Lexicon-derived relationships.** These relationships, generated by MELIS, represent complex relationships between meanings of terms annotated with lexical senses. These relationships may be inferred not only from lexical knowledge (e.g. by querying WordNet for relationships across senses), but also from background knowledge (e.g. domain ontologies) which are available at the time of the annotation. As we will say later (section 2), at any step MELIS can (re)use any piece of ontology generated by the current extraction process as a source of domain knowledge to incrementally refine the extraction of new relationships.
- **Designer-supplied relationships.** To capture specific domain knowledge, designers can supply new relationships directly.
- **Inferred relationships.** MOMIS exploits description logic techniques from ODB-Tools (Beneventano et al., 1997) to infer new relationships.

GVV generation The Global Virtual View (GVV) consists of a set of classes (called Global Classes), plus mappings to connect the global attributes of each global class and the local sources attributes. Such a view conceptualizes the underlying domain; you can think of it as an ontology describing the sources involved.

GVV and local sources annotation MOMIS automatically proposes a name and meanings for each global class of a GVV (Beneventano et al., 2003) Names and meanings have to be confirmed by the ontology designer. Local sources are conceptually annotated according to the created GVV.

4 Testing MOMIS+MELIS on a real domain

We tested MOMIS integrated with MELIS by building an ontology of a set of data-intensive websites[7] containing data related to the touristic domain (see figure 3), which have been wrapped and data from them have been structured and stored into four relational databases off-line available. The main classes of these sources are: hotel (of the "venere" database), restaurant ("touring" database), camping (guidaC database) and bedandbreakfast (BB database).

As discussed before, the incremental annotation process starts with the annotation of parts of the data sources, i.e. for each source element the ontology designer selects one or more corresponding WordNet synsets. For example, WordNet tells us that "hotel" and "restaurant" are siblings (i.e. they have a common direct hypernym); that "hotel" , "house", "restaurant" are direct hyponyms of "building"; that "bed and breakfast" is an hyponym of "building"; and that the closest hypernym that "campsite" and "building" share is "physical object", a top level synset in WordNet. As the last relationship does not allow finding lexical connections between "camping" and the other classes, we used the WNEditor to add a direct relationships between "campsite" and the hierarchy of "building".

Notice that the annotation process is a critical process: by annotating the source element "camping" as the WordNet synset "camping" a mistake would be generated because it means "the act of encamping". The correct synset for camping is "campsite", i.e. "the site where people can pitch a tent". Moreover, in order to test all the implemented heuristics,"hotel" has been annotated as its hypernym: "building".

The annotated schema is then given both as input and as the reference ontology to CtxMatch2.0. The tool starts the meaning elicitation process and produces a set of inferred lexical annotations of the schema elements. The resulting annotated schema is shown to the designer, who may validate and extend the annotation produced by CtxMatch2.0 and, eventually, restart the process using the updated annotated schema as reference ontology.

Figure 4 illustrates the results of a sample test of incremental annotation on one of our schemas. It shows the annotations manually provided by the ontology designer, a fraction of the new annotations generated after a first run of MELIS, and the additional annotations generated after a second run, when the outcome of the first run was provided as additional background knowledge in input; the numbers on the arrows refer to the heuristic rule which was used to generate the annotation. Notationally, a square near a class/attribute means that the element was manually annotated, a circle means that the element was automatically annotated after the first run, and a rhombus that it was incrementally annotated after the second run.

- Rule 1: the attribute "identifier" of the class "facility" in the source "VENERE" is annotated as "identifier" of the class "facility" in the source "BB" since both the classes are annotated with the same synset.
- Rule 2: because of the hyponym relationships generated by the annotations of the classes "hotel", "campsite", "bed and breakfast" and "building", the attribute "city" of the class "building" in the source "VENERE" produces the annotation of the same attribute in the sources "BB", "touring", "guidaC".
- Rule 3: because of the hypernym relation relationships generated by the annotations of "building" and "bed and breakfast", the attribute "identifier" of the class "bed_and_breakfast" in the source "BB" generates the annotation of the same attribute in the source "VENERE". By executing a second run of the MELIS process, the attribute "identifier" on the class "building" generates the annotation of the same attribute on the classes "campsite" and "restaurant" of the sources "guidaC" and "touring" (application of the heuristic rule 2).
- Rule 4: because of the new relationship introduced in WordNet, "campsite" is a sister term of "restaurant". Consequently, the attribute "locality" is annotated in the same way in the sources "guidaC" and "touring".
- Rule 5: in the source "VENERE" the class "map" has a foreign key: the attribute "url" that references to the class "hotel". Because of this relationship joins with hierarchical relationships "hotel", "campsite", "bed and breakfast" and "building", the annotation of attribute "url" of the class

[7] http://www.bbitalia.it, http://www.guidacampeggi.com, http://www.venere.com, http://www.touringclub.com

Figure 3: Sources used for evaluating MELIS

"map", applying Rule 3 and Rule 5, generates the same annotation for "url" in the classes "campsite", "bed_and_breakfast" and "restaurant" of the other sources.

Notice that heuristic 6 and 7 are not exploited in this example. Such rules may be exploited in nested structures as hierarchies, and they may not be applied in flat structures as relational databases.

The results are highly dependent on the annotation manually provided by the user as MELIS input . For this reason, it is not meaningful to give any evaluation in terms of number of new annotations discovered. Concerning the evaluation of the new annotations generated, our experience highlights that all of the new annotated elements have a correct meaning w.r.t. WordNet.

5 Conclusion and future work

In this paper we presented MELIS, a method and tool for incrementally annotating data sources according to a lexical database (WordNet in our approach). MELIS exploits the annotation of a subset of source elements to infer annotations for the remaining source elements, this way improving the activity of manual annotation.

MELIS is based on the integration and the extension of the lexical annotation module of the MOMIS-Ontology Builder (Benassi et al., 2004) and some components from CTXMATCH2.0, a tool for eliciting meaning and matching pairs of nodes in heterogeneous schemas, using an explicit and formal representation of their meaning (Bouquet et al., 2005; Bouquet et al., 2006). CTXMATCH2.0 was extended with respect to (Bouquet et al., 2005; Bouquet et al., 2006) with a set of heuristic rules to generate new annotations on the basis of the knowledge provided by a given set of annotations; *WNEditor* was modified in order to jointly work with CTXMATCH2.0, by providing a customized lexical database.

We experimented MELIS in conjunction with the MOMIS system in order to improve the MOMIS methodology for semi–automatically creating a domain ontology from a set of data sources. The first results show that MELIS and MOMIS working in conjunction are an effective tool for creating a domain ontology. The testing was performed within the WISDOM project[8], for creating an ontology from several data-intensive websites about hotels and restaurants.

As we noticed in the introduction, MELIS can be used to provide valuable input not only for ontology learning, but also for ontology matching tools. Here we want to notice that these tools can greatly benefit from the integration of MELIS, as MELIS provides a highly accurate collection of lexical annotations which can be exploited in the matching phase.

Future work on MELIS will be addressed on improving the annotation technique in order to deal with compound terms (like "full professor", "table leg", "football team"). Compound terms do not appear in any lexical database, unless they form a stable compound (e.g."station wagon"). Their annotation is therefore more difficult, as the choice of the right lexical meaning often depends on determining the relationship between terms.

Moreover, we will introduce in MELIS more accurate stemming techniques in order to improve the matching among input terms and the words of the lexical reference database. Finally, we are developing a methodology for building and sharing among a community new lexical database entries, e.g. by establishing how and when a new noun/meaning can be "promoted" to be part of the common lexical reference.

[8] http://www.dbgroup.unimo.it/wisdom

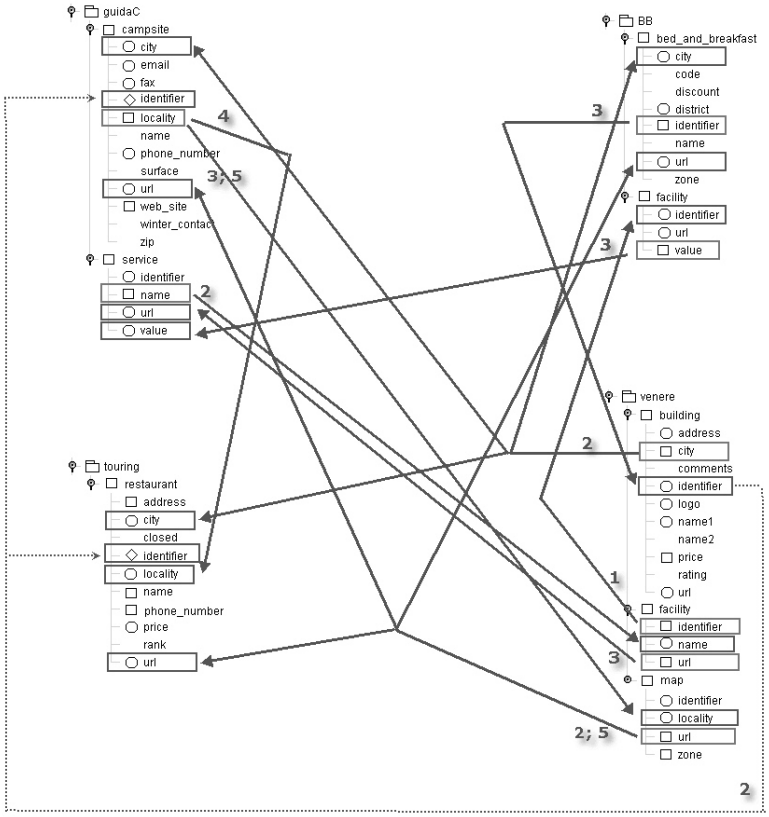

Figure 4: Annotations generated with MELIS

REFERENCES

Bechhofer, S., Carr, L., Goble, C. A., Kampa, S., and Miles-Board, T. (2002). The semantics of semantic annotation. In Meersman, R. and Tari, Z., editors, *CoopIS/DOA/ODBASE*, volume 2519 of *Lecture Notes in Computer Science*, pages 1152–1167. Springer.

Benassi, R., Bergamaschi, S., Fergnani, A., and Miselli, D. (2004). Extending a lexicon ontology for intelligent information integration. In de Mántaras, R. L. and Saitta, L., editors, *ECAI*, pages 278–282. IOS Press.

Beneventano, D., Bergamaschi, S., Guerra, F., and Vincini, M. (2003). Synthesizing an integrated ontology. *IEEE Internet Computing*, pages 42–51.

Beneventano, D., Bergamaschi, S., Sartori, C., and Vincini, M. (1997). ODB-QOPTIMIZER: A tool for semantic query optimization in oodb. In *Int. Conference on Data Engineering - ICDE97*. http://www.dbgroup.unimo.it.

Bouquet, P., Serafini, L., and Zanobini, S. (2005). Peer-to-peer semantic coordination. *Journal of Web Semantics*, 2(1).

Bouquet, P., Serafini, L., Zanobini, S., and Sceffer, S. (2006). Bootstrapping semantics on the web: meaning elicitation from schemas. In *WWW2006 Conference Proceedings*. W3C.

Gangemi, A., Navigli, R., and Velardi, P. (2003). The ontowordnet project: Extension and axiomatization of conceptual relations in wordnet. In Meersman, R., Tari, Z., and Schmidt, D. C., editors, *CoopIS/DOA/ODBASE*, volume 2888 of *Lecture Notes in Computer Science*, pages 820–838. Springer.

Handschuh, S., Staab, S., and Volz, R. (2003). On deep annotation. In *WWW*, pages 431–438.

Montoyo, A., Palomar, M., and Rigau, G. (2001). Wordnet enrichment with classification systems. In *In Proc. of WordNet and Other Lexical Resources: Applications, Extensions and Customisations Workshop, NAACL-01*, pages 101–106, Carnegie Mellon Univ., Pittsburgh, USA.

Pazienza, M. T. and Stellato, A. (2006). An open and scalable framework for enriching ontologies with natural language content. In Ali, M. and Dapoigny, R., editors, *IEA/AIE*, volume 4031 of *Lecture Notes in Computer Science*, pages 990–999. Springer.

Staab, S., Maedche, A., and Handschuh, S. (2001). An annotation framework for the semantic web. In *S. Ishizaki (ed.), Proc. of The First International Workshop on MultiMedia Annotation. January. 30 - 31. Tokyo, Japan*.