Lexical Knowledge Extraction: an Effective Approach to Schema and Ontology Matching

Laura Po, Serena Sorrentino, Sonia Bergamaschi and Domenico Beneventano
University of Modena and Reggio Emilia, Modena, Italy
laura.po@unimore.it
serena.sorrentino@unimore.it
sonia.bergamaschi@unimore.it
domenico.beneventano@unimore.it

Abstract: This paper's aim is to examine what role Lexical Knowledge Extraction plays in data integration as well as ontology engineering.
Data integration is the problem of combining data residing at distributed heterogeneous sources, and providing the user with a unified view of these data; a common and important scenario in data integration are structured or semi-structure data sources described by a schema.
Ontology engineering is a subfield of knowledge engineering that studies the methodologies for building and maintaining ontologies. Ontology engineering offers a direction towards solving the interoperability problems brought about by semantic obstacles, such as the obstacles related to the definitions of business terms and software classes.
In these contexts where users are confronted with heterogeneous information it is crucial the support of matching techniques.
Matching techniques aim at finding correspondences between semantically related entities of different schemata/ontologies.
Several matching techniques have been proposed in the literature based on different approaches, often derived from other fields, such as text similarity, graph comparison and machine learning.
This paper proposes a matching technique based on Lexical Knowledge Extraction: first, an Automatic Lexical Annotation of schemata/ontologies is performed, then lexical relationships are extracted based on such annotations.
Lexical Annotation is a piece of information added in a document (book, online record, video, or other data), that refers to a semantic resource such as WordNet. Each annotation has the property to own one or more lexical descriptions.
Lexical annotation is performed by the Probabilistic Word Sense Disambiguation (PWSD) method that combines several disambiguation algorithms.
Our hypothesis is that performing lexical annotation of elements (e.g. classes and properties/attributes) of schemata/ontologies makes the system able to automatically extract the lexical knowledge that is implicit in a schema/ontology and then to derive lexical relationships between the elements of a schema/ontology or among elements of different schemata/ontologies.
The effectiveness of the method presented in this paper has been proven within the data integration system MOMIS (Beneventano D. et al., 2003).

Keywords: disambiguation, lexical annotation, probabilistic lexical relationships, ontology engineering, data integration, ontology matching

1. Introduction

In the context of data integration and ontology engineering, matching has been recognised as a plausible solution for a long time (Euzenat and Shvaiko, 2007).
Ontology engineering has to deal with multiple, distributed and evolving ontologies. Ontology heterogeneity may be first faced while designing an ontology for a domain of interest; in this phase we have to integrate different ontologies for enforcing reuse and interconnecting various relevant resources.
Data integration is the problem of combining data residing at different sources, and providing the user with a unified view of these data (Lenzerini, 2002)The core of data integration is solving the correspondences (find the right matches) among elements from different data sources schemata. This is the reason that explains why data integration is one of the oldest applications where matching techniques have been applied (Batini et al., 1986).
In most real world applications, elements (e.g. classes and properties) of schemata/ontologies are labelled by natural language expressions. In our opinion, the crucial reason for this aspect is that natural language labels provide a rich connection between formal objects (e.g. classes and properties) and their intended meanings. The intuition is that grasping the intended interpretation of a element requires not only to understand the modelling logic behind structuring information (i.e the

structural relationships among schema elements), but also knowledge about the meaning behind the names denoting schemata/ontologies elements (i.e. recognize how the data are “labelled”).
In other words, a data source can be viewed as a collection of formal constraints between elements, whose intended meanings also depends on lexical knowledge.
Annotation becomes, thus, crucial to understand the meaning of schemata/ontologies.
Lexical annotation (i.e. the operation of associating an element of a lexical reference database to each source elements) is a difficult task, and making it accurate may require a heavy user involvement for several reasons:

- *coverage*: a complete lexical database including all possible terms does not exist. WordNet (WN), for example, contains a very large number of general terms, but does not cover specialized domains, whereas specialized lexical databases tend to disregard general terms;
- *polysemy*: in natural language, many terms are polysemous, namely may have many possible meanings. The choice of the specific meaning associated to the term is context dependent, and therefore this choice (called Word Sense Disambiguation in Natural Language Processing) is very difficult to automate;
- *integration*: a standard model/language for describing lexical databases does not exist. Consequently, it is difficult to integrate different lexical resources.

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 (http://annotation.semanticweb.org/tools/).

In this paper we present a flexible method to perform the lexical annotation of structured and semi-structured data sources. The method accomplish two important tasks: the source lexical annotation process and the discovery of lexical relationships among elements from the different data sources.
One key aspect for lexical annotation is that it has been more and more crucial that annotation process is performed automatically. Manual annotation is a time-consuming task, moreover it is difficult for a user to have a overall view of the sources and understand the context in which a term has been posed. While for small sources a user can be able to perform a good annotation, this task is harder for large sources. Automatic techniques do not have any difficulty to be applied on a large scale.
We proposed the PWSD method that automatically annotates source elements and associates to any annotation a probability value that indicates the reliability level of the annotation. PWSD is based on a probabilistic combination of different WSD (Word Sense Disambiguation) algorithms. After this task of probabilistic lexical annotation, it is possible to automatically extract probabilistic lexical relationships across elements of different schemata/ontologies, on the basis of relationships defined among meanings in the lexical database (WordNet in our case).
The novelty of our method relies on the completely automatic annotation of sources elements. In particular, our method associates to each annotation a probability value that represents the reliability level of the annotation. As a consequence, also the extracted lexical relationships are expressed in a probabilistic way.
This paper is organized as follows: in section 2 we introduce the definitions of schema and ontology matching, then, in section 3, we introduce the definition of Lexical Annotation.
Section 4 depict the method to perform lexical knowledge extraction on schemata or ontologies. In section 6 we sketch out some conclusion and future work.

2. Schema and Ontology Matching

Matching techniques aim at finding correspondences between semantically related entities of different schemata/ontologies. Several matching techniques have been proposed in the literature based on different approaches. In this section, after an introduction to the notions of ontology matching and schema matching, the definition of *lexical relationships* is given.

An ontology is an explicit specification of a conceptualization (Gruber, 1993) . An ontology defines a set of representational primitives with which to model a domain of knowledge or discourse. An ontology provides a shared vocabulary, which can be used to model a domain that is, the type of objects and/or concepts that exist, and their properties and relations. Ontologies are used in artificial intelligence, the Semantic Web, software engineering, biomedical informatics, library science, and information architecture as a form of knowledge representation about the world or some part of it. Ontologies are used to reason about the properties of that domain, and may be used to define the domain.
The ontology matching process, for two separate and autonomous ontologies, O1 and O2, consists of the following steps:

Step 1: Finding corresponding entities in ontologies O1 and O2;
Step 2: Representing the found correspondences and using it to achieve some goal.
For Step 1, the main ontology entities that can be considered, when finding correspondences between ontologies O1 and O2, are: classes (concepts), individuals (instances), and properties (relations). For Step 2, for using the found correspondences, they need to be represented in a suitable format.
The goals (i.e. usages) of ontology mapping determine, what candidates to consider, when we are finding the correspondences and determine also how to represent the correspondences.

A schema describes the structure of the data stored in an information source. For structured sources (e.g. relational database and object-oriented database) a schema is available while for semi-structured sources (e.g. XML files and HTML pages) a schema can be extracted by extraction suitable techniques as described in (Abiteboul et al., 2000).
Schema matching is the task of identifying syntactic and semantic correspondences between different schemata. Solving such match problems is of key importance to service interoperability and data integration in numerous application domains.
A large volume of literature has been devoted to the field of Schema matching (e.g. (Rahm and Bernstein 2001))

Schemas and ontologies provide a vocabulary of terms that describes a domain of interest; schemas and ontologies constrain the meaning of terms used in the vocabulary. Then schema matching and ontology matching present commonalities and techniques developed for both problems are of a mutual benefit.
The matching process we propose is the lexical knowledge extraction apply either to schemata or ontologies. It also permits to match a schema and an ontology, that is a relevant activity in database applications since a number of important database problems have been shown to have improved solutions by using an ontology to provide *precise semantics* for a database schema (An et al., 2006); these include federated databases, data warehousing, and information integration. Semantics of the data is captured by some kind of *semantic correspondences* between the database schema and the ontology.

As in (An et al., 2006), we abstract from the specific language for describing schemata or ontologies and we use a generic conceptual modeling language (CML), which contains *common* aspects of most semantic data models - such as ODL_{I3} we use in our data integration system, MOMIS - and ontology languages such as OWL. Specifically, the language allows the representation of *classes* (or concepts) i.e. unary predicates over individuals, *relationships* (or object properties) i.e. binary predicates relating individuals, and *attributes* (or datatype properties) i.e. binary predicates relating individuals with values such as integers and strings; classes are organized in the familiar *is-a* hierarchy.

In the sequel, we use T to denote a schema/ontology prescribed by the generic CML.
Classes, relationships and attributes are called entities and will be denoted by E; the name of an entity E will be called *term* and will be denoted by E.N; the matching process we propose is to identify correspondences between entities Ei on the basis of the meanings of their names Ei.N. These correspondences are called *lexical relationships* and are defined as follows:

- SYN: (Synonym) E1 SYN E2 iff E1.N is synonym of E2.N;
- BT: (Broader Term) E1 BT E2 iff E1.N is an hypernym of E2.N (the opposite of BT is NT (Narrower Term));
- RT: (Related Term) E1 RT E2 iff E1.N is a holonym or meronym of E2.N.

In section 4.2 we will discuss how these relationships can be automatically extract from schemata/ ontologies which are annotated with respect to a lexical resource.

3. Lexical Annotation

In literature we found a lot of definition of annotation in different application context.
An *Annotation*, in general, is a piece of information added in a book, document, online record, video, or other data (an intuitive example of annotation are the notes about the quality of a document written on the sheets of a draft document by a reader).
In the field of Semantic Web, annotation becomes a set of instantiations related to an ontology and referring to an HTML document (Handschuh et al., 2003) . This annotation is called *Metadata Annotation* or *Ontology-based Metadata Annotation*.

Natural language applications, such as information extraction and machine translation, require a certain level of semantic analysis. An important part of this process is semantic tagging (Buitelaar and Declerck, 2003): the annotation of each content word with a semantic category. The activity of semantic tagging refers to the activity of annotating text documents with a tags set defined on the ontology. There are different semantic resources that are available to be used in semantic tagging. Starting from metadata annotation and sense tagging we have derived and defined the concept of lexical annotation.

Definition - Lexical Annotation
Lexical Annotation is a particular kind of Metadata Annotation that refers to a semantic resource. Each annotation has the property to own one or more lexical descriptions.

Lexical Annotation differs from the *Ontology-based Metadata Annotation* where we annotate w.r.t. an ontology and it is not mandatory that an ontology class has a lexical description.
Semantic resources can be dictionaries, thesauri, and semantic networks, all of them express, either implicitly or explicitly, a general ontology of the world or of more specific domains, such as medicine.

We perform lexical annotation, by exploiting WSD techniques, w.r.t. the semantic resource WN (Fellbaum and Miller, 1998). WN is a large lexical database of English where nouns, verbs, adjectives and adverbs are grouped into sets of meanings (called synsets), each expressing a distinct concept. Synsets are interlinked in an hierarchy thesaurus by means of semantic and lexical relations. Its latest version, WN 3.0, contains about 155,000 terms organized in over 117,000 synsets. In the following, we denote the synset identifier for a term t as $t_{\#i}$, where i indicates the i^{th} WN synset for the term t (e.g. “book#1” indicates the first WN synset for the term “book”).

We choose WN as semantic resource for three main reasons: (1) WN has been proven to be very useful in the WSD context (Novischi, 2004); (2) the use of a well-known and shared thesaurus as WN provides a reliable set of meanings and allows to share with others the result of the annotation process; (3) the fundamental peculiarity of a thesaurus as WN is the presence of a wide network of relationships between terms and meanings.
However, our method can exploit other semantic resources providing a wide network of semantic relationships between meanings.

The lexical annotation of the sources (textual, structured or semi-structured) aims to solve the semantic differences among different data representations.
Lexical annotation is a critical task to develop smart methods for matching discovery, therefore, it should not come as a surprise that a large number of tools includes some lexical resource (mainly WN, available at http://wordnet.princeton.edu) as a component, and uses it in some intermediate step to annotate schema/ontology entities.
To perform lexical annotation, we need to explore the area of Word Sense Disambiguation (WSD). As it is described in (Ide and Veronis, 1998) the WSD task involves two steps: (1) the determination of all the different senses for every word under consideration; and (2) a mean to assign to each occurrence of a word its appropriate senses. The most recent works on WSD rely on predefined senses for step (1), including: a list of senses such as those found in semantic resources as dictionaries or thesauri.
The disadvantage in using a thesaurus (as WN) is that it does not cover with the same detail different domains of knowledge. Some terms may not be present or, conversely, other terms may have many associated and related meanings. These considerations and the first tests made led to the need of expanding the thesaurus with more specific terms (this can be easily done using the MOMIS component, called *WNEditor*, which allows adding new terms and linking them within WN (Benassi et al., 2004)). On the other hand, when a term have many associated and related meanings, we need to overcome the usual disambiguation approach and relate the term to multiple meanings: i.e. to union of the meanings associated to it. Even Resnik and Yarowsky (2000) ratify that there are common cases where several fine-grained senses may be correct.
For these reason, all the WSD methods, we use in PWSD, may associate more than one meaning to a term and is, thus, differs from the traditional approaches.
Once we have performed the annotation task, the elements of the sources have been enriched of new information: their meanings. These information are not unrelated, but they are connected in a lexical network (the relationships among WN synsets). After the annotation task, we found new lexical relationships across elements of different data sources.

4. Lexical knowledge extraction

The extraction of lexical knowledge from data sources, is based on the application of the PWSD method. During this phase, PWSD interacts with the lexical resource WN extended with WND (WordNet Domains). WND can be considered an extended version of WN, in which synsets have been annotated with one or more domain labels. WND has been proven a useful resource for WSD. In fact, it has been used in different combined WSD algorithm as presented in (Bergamaschi S. et al., 2007; Gliozzo A. M. et al. 2005). PWSD supplies a set of probabilistic annotations of the source terms: from these annotations probabilistic lexical relationships among source terms can be derived, as we will discuss in section 4.2.

Definition - Probabilistic Lexical Annotation

Let T be a schema and $E \in T$ be an entity of T. The probabilistic lexical annotation of E is the triple (T; E; A_E), where $A_E = \{a_1;\ldots; a_k\}$ is the set of annotation associated to E. In particular, a_i is defined as the couple ($N_{\#i}$; P), where

- $N_{\#i}$ is a synset for the term E.N w.r.t. a lexical resource (as WordNet)
- P is the probability value assigned; this probability indicates how well the meaning $N_{\#i}$ represented the term E.N.

Definition - Ordinary Lexical Annotation

An ordinary annotation for an entity $E \in T$ is a probabilistic lexical annotation where the probability value assigned to a set of meanings for a term E.N is equal to “1”.

An example of an ordinary lexical annotation could be a manual annotation (a user manually chooses one or more meanings to disambiguate a term), or an annotation assigned to a monosemous term. A monosemous term is described by a unique WN synset, therefore the disam<biguation is certain for this term.

4.1 The PWSD method

The PWSD method, we propose, is based on a probabilistic combination of different WSD algorithms.
PWSD allows to automatically “lexically annotate” a set of sources w.r.t a lexical resource (in our case WN). Given a term (i.e a schema/ontology element), a WSD (Word Sense Disambiguation) algorithm associates to it a set of meanings (synsets/senses) not necessarily orthogonal or mutually exclusive (Resnik and Yarowsky, 2000).
The performance of an annotation tool can be higher when a number of algorithms are used together (Stevenson and Wilks, 2001).To maximize annotation accuracy, we employ a variety of WSD algorithms we developed (Beneventano, et al., 2008), where each algorithm is associated with its own reliability. All the algorithms produce probability distributions on meanings of annotated terms.
The probability value of a term annotation derives from the reliability of the algorithm and the number of synsets chosen.
The use of different WSD algorithms leads to an epistemic uncertainty, i.e. the type of uncertainty which results from the lack of knowledge about a system. As a matter of fact, not every algorithm is able to disambiguate each term. In addition, each algorithm may be appropriate to certain situations, so its behaviour is not 100% trustworthy.
Looking for a flexible method in order to combine a variable number of algorithms and their outputs, we found out the Dempster-Shafer theory. The theory deals with the so-called frame of discernment, in our case, the set of all possible meanings for the term under consideration, and its power set, which is the set of all the possible subsets of the set of possible meanings.
We derive the belief mass function from the output and the reliability of the WSD algorithms. Using the Dempster's rule of combination we combine the output of the WSD algorithms and determine with which probability each source element can be associated to its meanings (Shafer, 1976).
With PWSD each term is disambiguated with the contribution of all the WSD algorithms.
PWSD has been implemented within MOMIS system (Beneventano et al., 2003),
but might be coupled with any data integration system or mapping tool (Po, 2009).
PWSD is as a modular framework: it is composed of a number of WSD algorithms but the addition or removal of algorithms is easy.

PWSD improves the annotation process by exploiting both structural and lexical knowledge of a set of data sources: a specific WSD algorithm, called Structural Disambiguation (Bergamaschi et al., 2008), exploits the structural relationships of a data source to annotate terms and to infer lexical relationships on the basis of WN; this algorithm explores WN to find a corresponding relationship when a structural relationship holds among two terms.

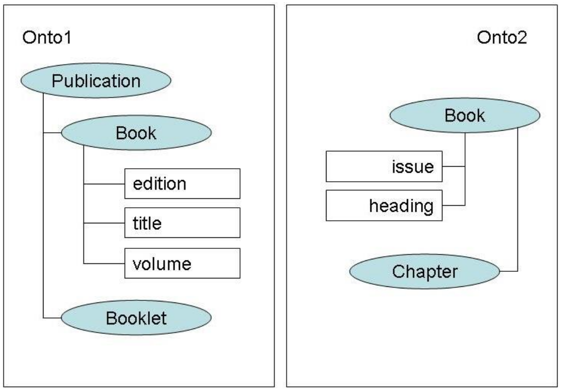
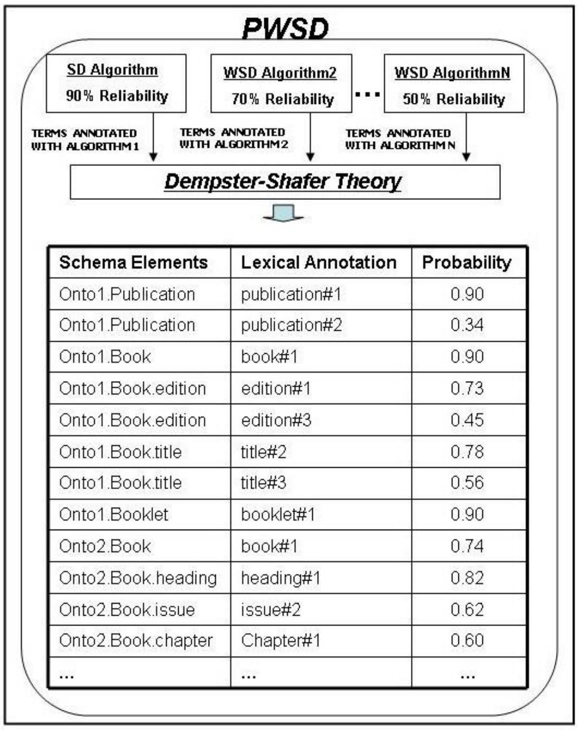


Figure 1 - Example of the application of PWSD: two bibliographic ontologies

As a case in point, let us consider the two ontologies shown in Figure 1.
By the use of PWSD, all the WSD algorithms can be apply on these sources and their outputs will be combined.
SD, for example, explores the WN relationships network in order to find out a meanings for the elements that are related by a structural relationships. In this example there is a ISA relationship between Onto1.Publication and Onto1.Book and between Onto1.Publication and Onto1.Booklet in the first Ontology.
Let us examine the annotation of the element Onto1.Book. The term book has eight different meanings in WN 3.0. SD provides book#1 as an annotation for the element because it finds out a hypernym relation between the meaning book#1 and publication#1 in WN. Suppose we have to combine SD with other algorithms that give different outputs (e.g. book#2….).
With PWSD we obtain a rate of confidence to be assigned to each possible meaning of the term under consideration. As each disambiguation algorithm does not provide a certain output, the combination will give rise to an uncertain output: the annotation book#1 assumes a probability of 0.95, book#2 a probability of 0.59.
At the end of the annotation process, PWSD returns the output shown in Figure 2, where each source element has been associated to one or more probabilistic annotations.



Schema Elements	Lexical Annotation	Probability
Onto1.Publication	publication#1	0.90
Onto1.Publication	publication#2	0.34
Onto1.Book	book#1	0.90
Onto1.Book.edition	edition#1	0.73
Onto1.Book.edition	edition#3	0.45
Onto1.Book.title	title#2	0.78
Onto1.Book.title	title#3	0.56
Onto1.Booklet	booklet#1	0.90
Onto2.Book	book#1	0.74
Onto2.Book.heading	heading#1	0.82
Onto2.Book.issue	issue#2	0.62
Onto2.Book.chapter	Chapter#1	0.60
...	...	...

Figure 2 - Example of the application of PWSD: annotation of two bibliographic ontologies

4.2 From the probabilistic Lexical Annotation to the discovery of probabilistic relationships

From the lexical annotation performed over sources terms it is possible to compute the lexicon relationships holding among terms.
Lexical relationships between local sources terms can be derived from the semantic relationships defined among WN meanings.
The application of PWSD method associates a term in a source to a set of probabilistic meanings. After the application of the method a term t is described by the meaning t#i with a certain probability. Because all the provided meanings are included in the lexical resource WN, each of them is located within a network of lexical relationships.
The WN relationships are the following:

- synonymy (similar relation);
- hyponymy (sub-name relation);
- hypernymy (super-name relation);
- holonymy (whole-name relation);
- meronymy (part-name relation);
- correlation (two terms share the same hypernym).

Once we have assigned a meaning $N_{\#i}$ to the element E1 and a meaning $M_{\#j}$ to the term E2 , and discovered than a WN relationship hold between the meanings, we derived one of the following *probabilistic lexical relationships:*

- < *SYN(E1, E2), P)* > if a synonymy WN relationship hold between $N_{\#i}$ and $M_{\#j}$
- < *BT(E1, E2), P)* > and < *NT(E1, E2), P)* > if $N_{\#i}$ is a hypernym of $M_{\#j}$ in the WN relationship network
- < *RT(E1, E2), P)* > if $N_{\#i}$ is a holonym/ meronym/ correlation of $M_{\#j}$ in the WN relationship network

Since the probability assigned to each meaning is a statistical independent event, the probability of occurrence of the lexical relationship between two terms is equal to the probability that the two events occur simultaneously. Thank to the formula of the join probability, the probability value assigned to the lexical relationships holding between the two terms depends on the probability value of the meaning under consideration for the terms:

$$P = P(N_{\#i}) \times P(M_{\#j})$$

Let us consider again the example of Figure 2, where each source element has been annotated.
Starting from these lexical annotations, exploiting the WN relationship network that links meaning to other meaning with a relation type, a set of probabilistic relationships is extracted, as shown in Figure 3, Figure 4.
For example, the annotation Book#1 of the term Onto2.Book and the annotation publication#1 of the term Onto1.Publication generate a NT relationship between the elements Onto2.Book and Onto1.Publication.

Lexical Relationships	Probability
Onto1.Publication BT Onto2.Book.issue	0.56
Onto1.Book SYN Onto2.Book	0.67
Onto1.Book.edition NT Onto2.Book.issue	0.28
Onto1.Book.title NT Onto2.Book.heading	0.44
Onto1.Booklet NT Onto2.Book	0.67
Onto2.Book NT Onto1.Publication	0.67
Onto2.Chapter RT Onto1.Book	0.54
Onto2.Chapter RT Onto1.Publication	0.54
...	...

Figure 3 - Example of the application of PWSD: lexical relationships among two bibliographic ontologies

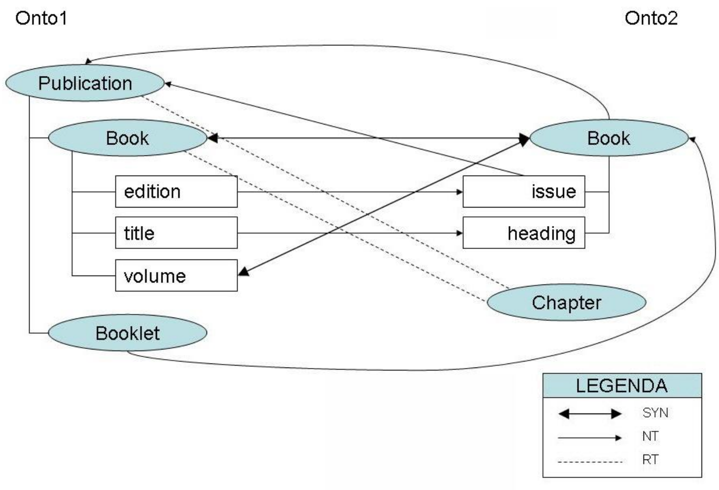


Figure 4 - Example of the application of PWSD: lexical relationships graphs of two bibliographic ontologies

5. Conclusion and Future Work

This paper proposed a matching technique based on Lexical Knowledge Extraction: first, an Automatic Lexical Annotation of schemata/ontologies is performed, then lexical relationships are extracted based on such annotations.
Lexical Annotation is a piece of information added in a document (book, online record, video, or other data), that refers to a semantic resource such as WordNet. Each annotation has the property to own one or more lexical descriptions.
Lexical annotation is automatically performed by the Probabilistic Word Sense Disambiguation (PWSD) method that combines several disambiguation algorithms.
We proved that by performing lexical annotation of elements (e.g. classes and properties/attributes) of schemata/ontologies we are able to automatically extract the lexical knowledge that is implicit in a schema/ontology and then to derive lexical relationships among elements of different schemata/ontologies.
The effectiveness of the method presented in this paper has been proven within the data integration system MOMIS (Beneventano D. et al., 2003).

Acknowledgements

This work was partially supported by MUR FIRB Network Peer for Business project (http://www.dbgroup.unimo.it/nep4b) and by the IST FP6 STREP project 2006 STASIS (http://www.dbgroup.unimo.it/stasis).

References

Abiteboul S. et al. (2000). Data on the Web: From relations to semi-structured data and XML. Data Management Systems. Morgan Kaufmann.

Agirre E. and Edmonds P., editors (2006) Word Sense Disambiguation: Algorithms and Applications, volume 33 of Text, Speech and Language Technology

An Y. et al. (2006) Discovering the semantics of relational tables through mappings. 4244:1-32

Batini C. et al. (1986). A comparative analysis of methodologies for database schema integration. *ACM Comput.Surv.*, 18(4): 323-364.

Benassi R. et al., (2004) Extending a lexicon ontology for intelligent information integration. In R. L. de Mantaras and L. Saitta, editors, *ECAI*: 278-282. IOS Press.

Beneventano D. et al., (2003) Synthesizing an integrated ontology. *IEEE Internet Computing*: 42-51.

Beneventano D., et al. (2008) Detailed design for building semantic peer. *Networked Peers for Business, Deliverable D.2.1, Final Version*: 52-57. Available at http://www.dbgroup.unimo.it/publication/d2 1.pdf.

Bergamaschi S. et al. (1999) Semantic integration of semi-structured and structured data sources. *SIGMOD Record*, 28(1):54-59.

Bergamaschi S. et al. (2007) Data source annotation in data integration systems. In Koch et al. editors. Proceedings of the 33rd International Conference on Very Large Data Bases, Austria ACM.

Bergamaschi S. et.al. (2008). Automatic annotation for mapping discovery in data integration systems. *SEBD*: 334-341.

Bernstein P. A. (2005). The many roles of meta data in data integration. In F. Ozcan editor, *SIGMOD Conference*, page 792. ACM.

Buitelaar P. and Declerck T. (2003). Linguistic annotation for the semantic web. In *Annotation for the Semantic Web*. IOS. Press.

Dong X. L. et al. (2007). Data integration with uncertainty. In C. Koch et al. editors, VLDB: 687-698. ACM.

Euzenat J. and Shvaiko P (2007): *Ontology matching*. Springer-Verlag,Heidelberg (DE).

Fellbaum C., Miller G. (1998) WordNet: An electronic lexical database. The MIT Press.

Gale WA, et al. (1992) One sense per discourse. In HLT '91: Proceedings of the workshop on Speech and Natural Language, Morristown, NJ, USA: Association for Computational Linguistics, 233–237.

Gliozzo A. M. et al. (2005) Domain kernels for word sense disambiguation. In ACL. The Association for Computer Linguistics.

Gruber T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199-220.

Handschuh S. et al. (2003) On deep annotation. In Proceedings of the 12th International World Wide Web Conference: 431-438.

Ide N. and Veronis J. (1998). Introduction to the special issue on word sense disambiguation: The state of the art. *Computational Linguistics*, 24(1):1-40.

Lenzerini M.. Data integration: A theoretical perspective. In L. Popa, editor,*PODS*: 233-246. ACM.

Navigli R. (2009): Word sense disambiguation: A survey. ACM Comput. Surv. 41(2):

Novischi, A. (2004): Combining methods for word sense disambiguation of wordnet glosses. In Barr, V., Markov, Z., eds.: FLAIRS Conference, AAAI Press.

Po L. (2009) Automatic Lexical Annotation: an effective technique for dynamic data integration. PhD Thesis. Available at http://www.dbgroup.unimo.it/~po/.

Rahm E. and Bernstein P. A. (2001). A Survey of Approaches to Automatic Schema Matching. VLDB Journal, 10:334–350.

Resnik P. and Yarowsky D. (2000) Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. *Natural LanguageEngineering*, 5(2):113-133.

Shafer G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

Stevenson M. and Wilks Y. (2001) The interaction of knowledge sources in word sense disambiguation. Computational Linguistics, 27(3):321-349.

Yarowsky D et al. (1993) One sense per collocation. In HLT '93: Proceedings of the workshop on Human Language Technology, Morristown, NJ, USA: Association for Computational Linguistics, 266–271.