Dealing with Uncertainty in Lexical Annotation

Sonia Bergamaschi, Laura Po, Serena Sorrentino and Alberto Corni[1]

Abstract: We present ALA, a tool for the automatic *lexical annotation* (i.e. annotation w.r.t. a thesaurus/lexical resource) of structured and semi-structured data sources and the discovery of probabilistic lexical relationships in a data integration environment. ALA performs automatic lexical annotation through the use of probabilistic annotations, i.e. an annotation is associated to a probability value. By performing probabilistic lexical annotation, we discover probabilistic inter-sources lexical relationships among schema elements. ALA extends the lexical annotation module of the MOMIS data integration system. However, it may be applied in general in the context of schema mapping discovery, ontology merging and data integration system and it is particularly suitable for performing "on-the-fly" data integration or probabilistic ontology matching.

1 Introduction

Traditional data integration systems are systems interconnecting a limited number of resources, which are relatively stable in time and have been typically built with complex and time-consuming design activities. As underling in [1], data-integration systems need to handle uncertainty at three levels: (1) on the semantic mappings between the data sources and the mediated schema, (2) on the keywords queries and (3) on the sources that may yield imprecise data. A powerful mean to discover mappings is the understanding of the meaning behind labels denoting schemata elements [2].

In this paper, we present ALA (Automatic Lexical Annotator), a tool that deals with the uncertain meaning of schema labels (thus, ALA handles uncertainty at level (1)). Using a probabilistic view of the meanings associated to a schema label, ALA performs automatic lexical annotation of the schema elements. This allows to discover probabilistic lexical relationships between heterogeneous data sources, which are collected in a Probabilistic Common Thesaurus (PCT). The idea of a probabilistic annotation is new in the field of data integration, although it is a well known approach in text disambiguation [3]. Lexical annotation associates a meaning (synset in WordNet[2]) or a set of meanings to a schema label. Probabilistic lexical annotation adds a probability value that indicates the reliability of the annotation.

[1] DII, Università di Modena e Reggio Emilia, Via Vignolese, 905, 41125 Modena, Italy
{firstname.lastname@unimore.it}
[2] http://wordnet.princeton.edu/

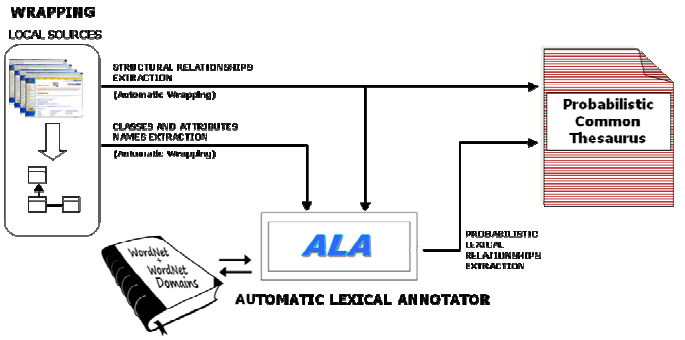

Figure 1 - ALA and the PCT

Several reasons have led us to use probabilistic annotation: (1) *multiple-annotations*: given a schema label, a WSD (Word Sense Disambiguation) algorithm associates to it a set of meanings not necessarily orthogonal or mutually exclusive; (2) *combined-techniques*: an ensemble of WSD algorithms of different nature overcomes the weaknesses of single approaches and maximizes annotation accuracy; (3) *uncertainty*: 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), not every algorithm is able to disambiguate each term, in addition, each algorithm may be appropriate to certain situations, so its behavior is not 100% trustworthy. ALA was developed within the MOMIS system [4], but might be coupled with any data integration system or mapping tool [5].

2 ALA Overview

ALA uses specialized software (wrappers) for logically converting data source schemata formats (relational, object, XML, XML-Schema) into the internal object language ODL_I^3 of the MOMIS system. Wrappers automatically extracts structural relationships from schemata and ODB-Tools [6], a description logic engine, infers new relationships by computing the transitive closure on the extracted relationships. Structural relationships are inserted in PCT with a probability value equals to 1. The structural relationships are:

- BT_{EXT} : t1 BT_{EXT} t2 iff extension(t2) $\subseteq$ extension(t1) (i.e. ISA, foreign key)
- SYN_{EXT} : t1 SYN_{EXT} t2 iff extension(t1) = extension(t2).

ALA provides a set of algorithms and operators to perform lexical annotation. From the scientific developer's perspective, ALA is a modular framework, which can easily be expanded. The implementation of new algorithms and operators is easy and intuitive. At

present ALA includes five different algorithms[3]: *Structural Disambiguation* algorithm examines terms that are related by a structural relationship (BT_{EXT} or SYN_{EXT}) and searches in the thesaurus (in our case WordNet) for a lexical relationship between the meanings associated to these terms; *WordNet Domains Disambiguation* algorithm tries to disambiguate terms by exploiting domains information supplied by WordNet Domains [7]; *Gloss Similarity* algorithm and *Iterative Gloss Similarity* algorithm are based on string similarity techniques; *WordNet first sense* heuristic rule selects the first WordNet meaning (that is the more used in English) for a term.

ALA assigns to each algorithm a reliability value (the default value of the reliability is the precision of the algorithm evaluated on a benchmark). The user can choose all or a subset of these algorithms and combine the algorithms outputs by using different operators. *Pipe* operator combines the annotation outputs of different algorithms provided in a given order. The pipe operator uses the output of the first algorithm and for the terms where no annotation is provided, executes the second algorithm and so on. With this operator, each term is disambiguated at most by a single algorithm. *Parallel* operator combines the annotation results from different algorithms by using the Dempster's rule of combination [8]. With the parallel operator, each term is disambiguated with the contribution of all the selected algorithms. *Threshold* operator filters out the annotations with a probability under a given value. Starting from the set of probabilistic annotations, ALA computes a probabilistic lexical relationship between two terms, if it exists a relationships between their meanings in the thesaurus. The lexical relationships are defined on the basis of WordNet relationships:

- *SYN*: defined between two terms that are synonymous;
- *BT*: (Broader Term) defined between two terms where an hypernym relationship holds between them (the opposite of *BT* is *NT* (Narrower Term));
- *RT*: (Related Term) defined between two terms when an holonym or meronym relationship holds between the terms.

The probability value assigned to a lexical relationship depends on the probability value of the meanings under consideration for each term. Thanks to the formula of the *join probability*, the probability value associated to a lexical relationship holding among the meanings $t_{\#i}$ and $s_{\#j}$ of terms t and s respectively, is defined as:

$$P(t\#i, s\#j) = P(t\#i) * P(s\#j) \quad (1)$$

3 Demonstration Content

We demonstrate as ALA, by exploiting both structural and lexical knowledge, provides a good quality probabilistic annotation drastically reducing human intervention and discovers probabilistic lexical relationships among schemata. For sake of simplicity, in the

[3] more details can be found at www.dbgroup.unimo.it/publication/d2_2.pdf

demo[4] we consider only three data sources from the benchmark 2008 of the OAEI project[5] but the process is scalable and can be performed on several scenarios, thus, the user can provide her or his own set of data sources (the sources may be expressed on XML, OWL, RDF or the main formats for DBMS). The demo starts with the extraction and conversion in ODL_I[3] of the schemata of the given set of data sources and the automatic extraction and inference of structural relationships. Then, the demo shows how the user may select among three different execution modalities: (1) *Default/Sequential* - the inexpert user does not set any parameters; algorithms are executed by using the pipe operator following the reliability order (or a manual order); (2) *Parallel* - the skilled user may select the algorithms to be applied; the parallel execution can be performed without/with threshold filtering; (3) *Formula* - the skilled user may combine algorithms and operators as she/he wishes, using the GUI or directly writing the formula.

ALA is an effective annotation analysis tool. As shown in the demo, through the GUI the user may have an estimation of the quality of the obtained annotations in terms of the number of annotated terms, the average probability of the annotations and the number of annotations per term. Thus, a user may easily determine the right combination of WSD algorithms to optimize the process. After the annotation, ALA computes the lexical relationships extraction: we demonstrate as the PCT is enriched with the discovered probabilistic lexical relationships.

References

[1] X. L. Dong, et al. Data integration with uncertainty. In VLDB, pages 687.698, 2007.

[2] F. Giunchiglia, P. Shvaiko, and M. Yatskevich. S-match: an algorithm and an implementation of semantic matching. In Semantic Interoperability and Integration, 2005.

[3] P. Resnik and D. Yarowsky. Distinguishing systems and distinguishing senses: new evaluation methods for word sense disambiguation. Natural Language Engineering, 2000.

[4] D. Beneventano, S. Bergamaschi, F. Guerra, and M. Vincini. Synthesizing an integrated ontology. IEEE Internet Computing, pages 42.51, Sep-Oct 2003.

[5] L. Po. Automatic Lexical Annotation: an effective technique for dynamic data integration. PhD Thesis, 2009. Available at http://www.dbgroup.unimo.it/ po/.

[6] D. Beneventano, S. Bergamaschi, and C. Sartori. Description logics for semantic query optimization in object-oriented database systems. ACM Trans. Database Syst., 2003.

[7] A. M. Gliozzo, C. Strapparava, and I. Dagan. Unsupervised and supervised exploitation of semantic domains in lexical disambiguation. Computer Speech & Language, 2004.

[8] G. Shafer. A Mathematical Theory of Evidence. Princeton University Press, 1976.

[4] http://www.dbgroup.unimo.it/ALA/ALATool.mp4
[5] http://oaei.ontologymatching.org/2008/