Improving Semantic Awareness of Knowledge-based Applications through Structural Disambiguation*

Federica Mandreoli[1], Riccardo Martoglia[1], and Enrico Ronchetti[1]

Università di Modena e Reggio Emilia
Dipartimento di Ingegneria dell'Informazione
41100 Modena, Italy
{fmandreoli,rmartoglia,eronchetti}@unimo.it

Abstract. In this paper, we summarize the features of the versatile disambiguation approach we recentlty presented in [8]. Its main aim is to make explicit the meaning of structure-based information such as XML schemas, XML document structures, web directories, and ontologies. It can be of support to the semantic-awareness of a wide range of applications, from schema matching and query rewriting to peer data management systems, from XML data clustering to ontology-based automatic annotation of web pages and query expansion. The effectiveness of the achieved results has been experimentally proved and is founded both on a flexible exploitation of the structure context, whose extraction can be tailored on the specific application needs, and of the information provided by commonly available thesauri such as WordNet. This work is partially supported by the Italian Council co-funded project WISDOM.

1 Introduction

In recent years, knowledge-based approaches, i.e. approaches which exploit the semantics of the information they access, are rapidly acquiring more and more importance in a wide range of application contexts. We refer to "hot" research topics, like schema matching and query rewriting [9], also in peer data management systems (PDMS) [6], XML data clustering and classification [12, 13] and ontology-based annotation of web pages and query expansion [3], all going in the direction of the Semantic Web [1]. In these contexts, most of the proposed approaches share a common basis: They focus on the structural properties of the accessed information, which are represented adopting XML or ontology-based data models, and their effectiveness is heavily dependent on knowing the right meaning of the employed terminology. Generally speaking, due to the ambiguity of natural languages, terms describing information usually have several meanings and making explicit the semantics of information goes through the tricky task of deriving from the context the most appropriate meanings. Fig. 1 shows the hierarchical representation of a portion of the categories offered by eBay,

* An extended version of this paper has been presented at ACM CIKM'05 [8]

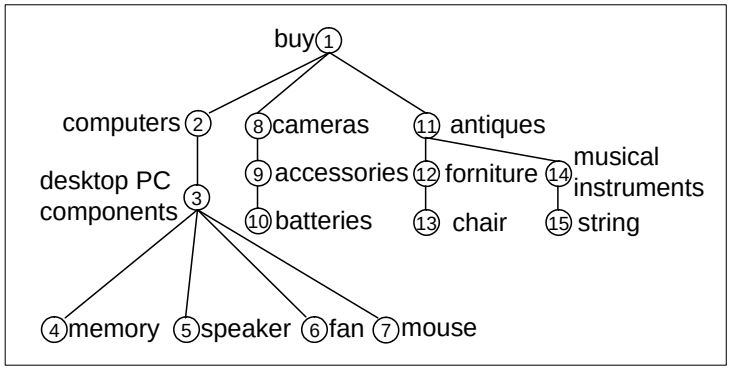

Fig. 1. A portion of the eBay categories.

one of the most famous online marketplaces (nodes are univocally identified by their pre-order values). It is an example of a typical tree-like structure-based information managed in the above mentioned contexts and which our approach is successfully able to disambiguate. It contains many polysemous words, from `string` to which WordNet [11], the most used commonly available vocabulary, associates 16 meanings, to `batteries` (11 meanings), `memory` (10 meanings), and so on. The information given by the surrounding nodes allows us to state, for instance, that `string` is a "stringed instrument played with a bow" and not a "linear sequence of symbols", and `batteries` are electronic devices and not a group of guns or whatever else.

For most of the knowledge-based approaches present in the literature, the problem of making explicit the meanings of words is usually demanded to human intervention and till now machine-based solutions have only been marginally addressed in the context of structure-based information. On the other hand, in the most cutting edge semantic-aware application contexts the role of humans is limited to that of user while, when some kind of human intervention is provided for, unassisted semantic annotation is a quite tedious task.

In this paper, we provide a summary of the main features of the approach we are working on (and whose current status we recently presented in [8]) for the disambiguation of graph-like structured information, mainly focusing on trees. It can be used to make explicit the meaning of a wide range of structure-based information, including XML schemas, the structures of XML documents, web directories, and ontologies. The rest of the paper is organized as follows: Section 2 presents an overview of our disambiguation approach. Experimental evaluation is provided in Section 3 and related works in Section 4. Finally, Section 5 concludes the paper.

2 Overview of the approach

In this section we present the functional architecture of a generic tree disambiguation service (see Fig. 2). The service takes in input structure-based infor-

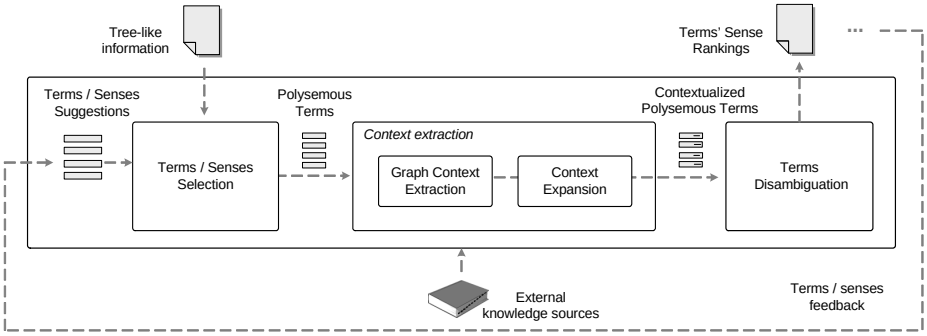

Fig. 2. The versatile structural disambiguation service.

mation like XML schemas, structures of XML documents and web directories and disambiguates the terms contained in each node's label using WordNet as external knowledge source. The outcome of the disambiguation process is a ranking of the plausible senses for each term. In this way, the system is able to support both the completely automatic semantic annotation whenever the top sense of the ranking is selected and the assisted one through a GUI that assists the user providing useful suggestions. The different components of the system are:

- **Terms/Senses Selection** component, which takes the label of each node N of the tree, extracts the contained terms (which can also be more than one as for instance `desktop PC components` of node 3 in Fig. 1) and associates each of these terms (t, N) with a list of senses $Senses(t, N) = [s_1, s_2, \ldots, s_k]$. In principle, such list is the complete list of senses provided by WordNet but it can also be a shrunk version suggested either by human or machine experts or as feedback of a previous disambiguation process.
- **Graph Context Extraction** component, which contextualizes each polysemous term (t, N) by extracting its *graph context* from the set of terms belonging to the reachable nodes. The nodes reachable by the term's node N are configurable by a crossing setting in which the maximum number of crossing and the allowed axes (ancestor/descendant/parent/child/sibling) are expressed. In the resulting graph context, each node is associated with a weight which is directly proportional to the term's closeness and which expresses its ability to influence the disambiguation.
- **Context Expansion** component, which expands the graph context with the nouns contained in the description and in the examples of each sense s in $Senses(t, N)$ and produces an *expanded context*. It is particularly useful when the graph context provides too little information.
- **Term Disambiguation** component, which finally disambiguates each term (t, N) with the associated senses $Senses(t, N)$ by using the previously extracted context. The result is a ranked version of $Senses(t, N)$ where each sense $s \in Senses(t, N)$ is associated with a confidence $\phi(s)$ in choosing s as a sense of (t, N).

The service can be also combined with an *interactive and/or automated feedback* to increase the quality of the disambiguation results.

The disambiguation algorithm we devised follows a relational information and knowledge-driven approach. Indeed, the context is not merely considered as a bag of words but other information such as their distance from the polysemous term to be disambiguated and semantic relations are also extracted. Moreover we use additional information provided by thesauri: The WordNet's hypernymy/hyponymy relationships among senses and the sense explanations and frequencies of use. The algorithm takes in input a term (t, N) to be disambiguated and produces a vector ϕ of confidences in choosing each of the senses in $Senses(t, N)$. In particular, given $Senses(t, N) = [s_1, s_2, \ldots, s_k]$, $\phi[i]$ is the confidence in choosing s_i as sense of (t, N). The senses showing the highest confidences will be the most recommended ones. The confidence vector ϕ tunes two contributions:

- the contribution of the context (that is composed by the contribution of the graph context and that of expanded context)
- the contribution of the frequency of sense use in English language

The contribution of the context is based on the similarity between terms and relies on the hypernymy/hyponymy hierarchy of WordNet; the intuition is: *the more similar two terms are, the more informative will be the most specific concept that subsumes them both* (common hypernym). Our measure is derived from the one proposed in [5]:

$$sim(t, t_c) = \begin{cases} -ln\frac{len(t,t_c)}{2 \cdot H} & \text{if } \exists \text{ a common hypernymy} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $len(t, t_c)$ is the minimum among the number of links connecting each sense in $Senses(t, N)$ and each sense in $Senses(t_c, N_c)$ and H is the height of the hypernymy hierarchy (in WordNet it is 16). The graph context contribution is computed by exploiting the similarity between term t and terms t_c in the graph context using Eq. 1.

Beside the contribution of the graph context, also the expanded context can be exploited in the disambiguation process. In this case, the main objective is to quantify the semantic correlation between the terms in the graph context and the nouns contained in the explanation and in the examples of each sense s in $Senses(t, N)$. The last contribution exploits the frequency of use of the senses in English language by means of a decay function. In particular, WordNet orders its list of senses $Senses(t, N)$ of each term t on the basis of the frequency of use (i.e. the first is the most common sense, etc.). We increment the confidence in choosing each sense s in $Senses(t, N)$ in a way which is inversely proportional to its position. Such an adjustment attempts to emulate the common sense of a human in choosing the right meaning of a noun when the context gives little help. For an in-depth description of the disambiguation algorithm see [8].

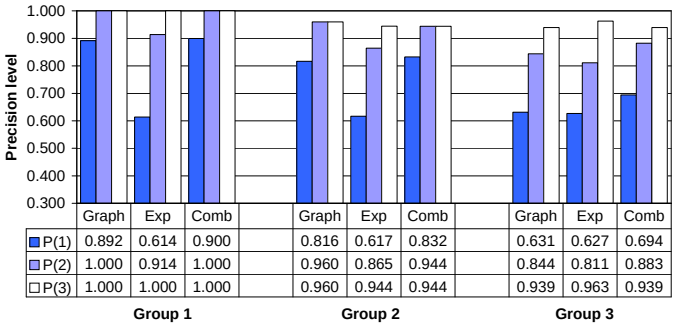

Fig. 3. Mean precision levels for the three groups

3 Experimental evaluation

In this section we present the most important results of an actual implementation of our disambiguation approach. For further results see [8]. Tests were conceived in order to show the behavior of our disambiguation approach in different scenarios. In particular we tested 3 groups of trees characterized by 2 dimensions of interest: *Specificity*, indicating how much a tree is contextualized in a particular scope, and *polysemy*, expressing how much the terms are ambiguous [8]. Group1 is characterized by a low specificity and a polysemy which increases along with the level of the tree; it is the case of web directories in which we usually find very different categories under the same root and a low polysemy at low levels and high polysemy at the leaf level (i.e. eBay's catalog). Group2 is characterized by a high specificity and a high polysemy, as for instance in the structure of the IMDb repository (`www.imdb.org`). Finally, Group3 is characterized by a high specificity and a low polysemy and contains representative XML schemas, such as DBLP digital library schema.

In our experiments we evaluated the performances of our disambiguation algorithm mainly in terms of effectiveness. Traditionally, word sense disambiguation (wsd) algorithms are evaluated in terms of precision and recall figures [4]. Recall parameter is not considered because its computation is usually based on frequent repetitions of the same terms in different documents, and we are not interested in evaluating the wsd quality from a single term perspective.

The disambiguation algorithm has first been tested on the entire collection of trees using the default graph context: all the terms in the tree. Fig. 3 shows the precision results for the disambiguation of the three groups. Three contributions are presented: The graph context one (Graph), the expanded context one (Exp) and the combined one (Comb). In general, precision P is the mean of the number of terms correctly disambiguated divided by the number of terms in the trees of

each group. Since we have at our disposal complete ranking results, we compute precision P(M) at different levels of quality, by considering the results up to the first M ranks: For instance, P(1) will be the percentage of terms in which the correct senses are at the top position of the ranking. Combination of graph context and expanded context contributions produces good P(1) precision levels of 90% and of over 83% for groups Group1 and Group2, respectively. Precision results for Group3 are lower (nearly 70%), due to the higher number and to the higher similarity between the senses of the involved terms (see Table 1 in [8] for detailed features of the involves trees); even in this difficult settings, the results are quite encouraging, particularly if we notice that P(2) is above 88%. As to the effectiveness of the context expansion, notice that its contribution alone (Exp) is generally very near to the graph context one, particularly in the complex Group3 setting, meaning a good efficacy of this approach too; further, in all the three cases the combination of the two contributions (Comb) produces better results than each of the contributions alone. For instance, the term `article` of DBLP schema is erroneously disambiguated as "a separate section of a legal document" using only the graph context and as "nonfictional prose forming an independent part of a publication" using also the expanded context.

One of the major strengths of our system is the versatility of being able to choose the crossing setting that is best suited to the tree characteristics. For instance, when the crossing setting is made up of the whole tree, the term `antique` of Fig.1 is not disambiguated as "an old piece of furniture or decorative object", but as "an elderly man" due to the presence of terms like `fan` and `speaker` that could have the meaning of "persons" rather than "objects". Notice that, using the eBay tree, the combined precision P(1) raises from 88% to 94% for a selected setting involving only ancestors, descendants and siblings. This behavior is typical of trees that gather very heterogeneous concepts like web directories. On the other hand, only by using the whole tree as the crossing setting in trees that have a very particular scope, for instance an IMDb tree schema on movies, terms like `episode` and `genre` are correctly disambiguated whereas a restricted crossing setting made of only ancestors and descendants provides wrong results. The IMDb combined precision drops from nearly 88% to 80% when only ancestors and descendant terms are kept. For more detailed results about the disambiguation process with different crossing settings see [8].

In general, the performed tests demonstrate that most of the term's senses are correctly assigned straightforwardly with the disambiguation (the mean precision level on the tested trees is generally over 80% [8]). Such good performance is obtained even when the graph context provides too little information, as in generic bibliographic schemas, thanks to the *context expansion* feature. To get even better results the user could choose to refine them by performing successive disambiguation runs; for this purpose he/she is able to deactivate/activate the influence of the different senses of the available context words on the disambiguation process through the GUI. Further, the flexibility of our approach allows the user to benefit from a completely *automated feedback*, where the results of the

first run are refined by automatically disabling the contributions of all but the top ranked X senses in the following runs.

4 Related work

Structural disambiguation is acknowledged as a very real and frequent problem for many semantic-aware applications. However, to our best knowledge, up to now it has only been partially considered in two contexts, schema matching and the XML data clustering, and few actual structural disambiguation approaches have recently been presented. In many schema matching approaches, the semantic closeness between nodes relies on syntactic approaches, such as simple string matching possibly considering its synonyms (e.g. [10, 7]). Also, a good number of statistical wsd approaches have been proposed in the matching context (e.g. [6]). However, they rely on additional data which may not always be available. As to the proper structural disambiguation approaches, in [13] the authors propose a technique for XML data clustering, where disambiguation is performed on the documents' tag names. The local context of a tag is captured as a bag of words containing the tag name itself, the textual content of the element and the text of the subordinate elements and then it is enlarged by including related words retrieved with WordNet. This context is then compared to the ones associated to the different WordNet senses of the term to be disambiguated by means of standard vector model techniques. In a similar scenario, the method proposed in [12] performs disambiguation by applying a shortest path algorithm on a weighted graph constructed on the terms in the path from each node to the root and on their related WordNet terms. For the graph construction, WordNet relations are navigated just one level. In a schema matching application, [2] presents a node disambiguation technique exploiting the hierarchical structure of a schema tree together with WordNet hierarchies. In order for this approach to be fully effective, the schema relations have to coincide, at least partially, with the WordNet ones.

Generalizing, our approach differs from the existing structural disambiguation approaches as it has not been conceived in a particular scenario but it is versatile enough to be applicable to different semantic-aware application contexts. It fully exploits the potentialities of the context of a node in a graph structure and its extraction is flexible enough to include relational information between the nodes and different kinds of relationships, such as ancestors, descendants or siblings. Further, we fully exploit WordNet hierarchies, and in particular the hypernym ones which are the most used for building effective relatedness measures between terms in free text wsd.

5 Conclusion

Our main aim was to provide a significant improvement to the semantic-awareness of a wide range of knowledge-based applications. In conclusion, the effectiveness of our approach is quite encouraging and induces us to continue in this direction.

In our future work, we will deeply analyse the ontology disambiguation problem, also by evaluating the performance on generic graphs, and the feedback process.

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 284(5), 2001.
2. P. Bouquet, L. Serafini, and S. Zanobini. Semantic coordination: a new approach and an application. In *Proc. of the 2nd ISWC Conference*, 2003.
3. P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proc. of the 13th WWW Conference*, 2004.
4. N. Ide and J. Veronis. Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art. *Computational Linguistics*, 24(1), 1998.
5. C. Leacock and M. Chodorow. Combining local context and WordNet similarity for word sense identification. In C. Fellbaum, editor, *WordNet: An electronic lexical database*. MIT Press, 1998.
6. J. Madhavan, P. A. Bernstein, A. Doan, and A. Y. Halevy. Corpus-based schema matching. In *Proc. of 21st ICDE Conference*, 2005.
7. J. Madhavan, P. A. Bernstein, and E. Rahm. Generic Schema Matching with Cupid. In *Proc. of the 27th VLDB Conference*, 2001.
8. F. Mandreoli, R. Martoglia, and E. Ronchetti. Versatile Structural Disambiguation for Semantic-aware Applications. In *Proc. of the 14th Conference on Information and Knowledge Management (CIKM)*, 2005.
9. F. Mandreoli, R. Martoglia, and P. Tiberio. Approximate Query Answering for a Heterogeneous XML Document Base. In *Proc. of the 5th WISE Conference*, 2004.
10. S. Melnik, H. Garcia-Molina, and E. Rahm. Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching. In *Proc. of the 18th ICDE*, 2002.
11. G. A. Miller. WordNet: A Lexical Database for English. *CACM*, 38(11), 1995.
12. A. Tagarelli and S. Greco. Clustering Transactional XML Data with Semantically-Enriched Content and Structural Features. In *Proc. of the 5th WISE Conference*, 2004.
13. M. Theobald, R. Schenkel, and G. Weikum. Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data. In *Proc. of the WebDB Workshop*, 2003.