STRIDER: a Versatile System for Structural Disambiguation*

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

DII, Università degli Studi di Modena e Reggio Emilia,
via Vignolese, 905/b - I 41100 Modena
⟨fmandreoli, rmartoglia, eronchetti⟩@unimo.it

Abstract. We present STRIDER[1], a versatile system for the disambiguation of structure-based information like XML schemas, structures of XML documents and web directories. The system performs high-quality fully-automated disambiguation by exploiting a novel and versatile structural disambiguation approach.

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 [2, 5], also in peer data management systems (PDMS), XML data clustering and classification [8] and ontology-based annotation of web pages and query expansion [1, 3], all going in the direction of the Semantic Web. 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. Fig. 1-a shows the hierarchical representation of a portion of the categories offered by eBay. 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 [6], 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.

In this paper we propose STRIDER, a system which could be of support to these kinds of approaches in overcoming the ambiguity of natural language, as it makes explicit the meanings of the words employed in tree-like structures. STRIDER exploits the novel versatile structural disambiguation approach we proposed in [4].

* This work is partially supported by the Italian Council co-funded project WISDOM.

[1] STRucture-based Information Disambiguation ExpeRt

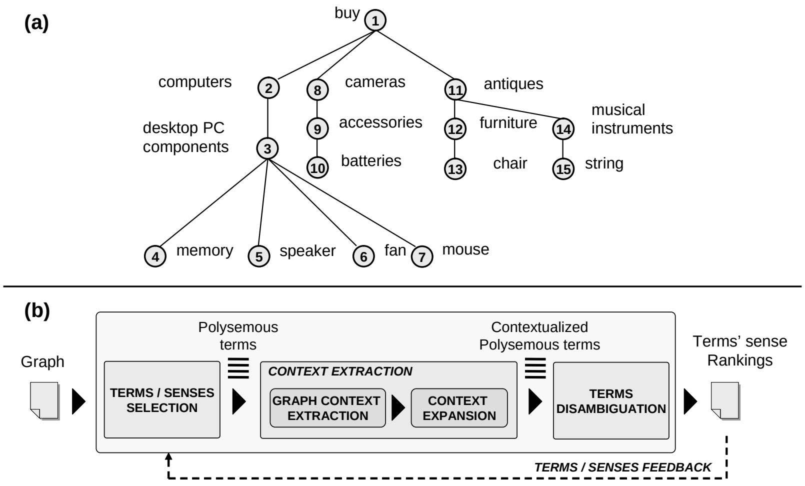
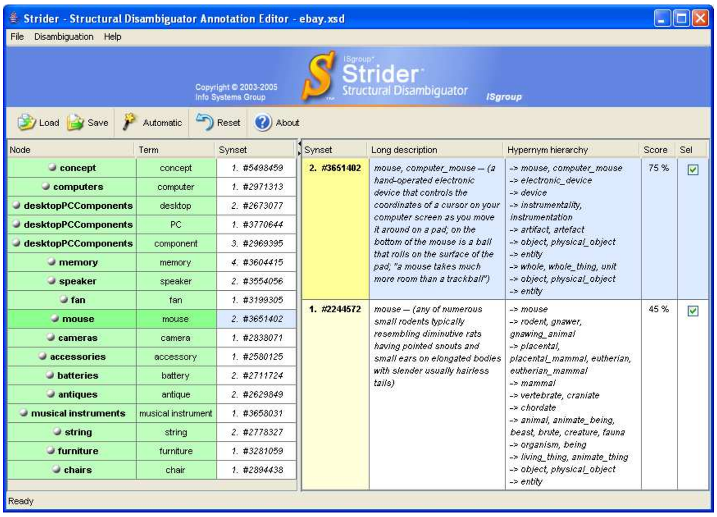


Fig. 1. (a) A portion of the eBay categories;(b) The complete STRIDER architecture

2 An overview of the STRIDER System

STRIDER is designed to perform effective disambiguation of tree-like structures. As shown in Fig. 1-b, which depicts the complete architecture of our system, STRIDER takes in input structure-based information 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 STRIDER system has the following features:

- automated extraction of terms from tree's nodes (**Terms/Senses Selection** component in Fig.1-b);
- high-quality and *fully-automated disambiguation* that:
  - is independent from training or additional data, which are not always available [7];
  - exploits a context which goes beyond the simple "bag of words" approach and preserves the information given by the hierarchy (*graph context*);
  - allows flexible extraction and full exploitation of the graph context according to the application needs (**Graph Context Extraction** component in Fig.1-b);
  - enriches the graph context by considering the *expanded context*, with additional information extracted from WordNet definitions and usage examples (**Context Expansion** component in Fig.1-b);
- *interactive and automated feedback* to increase the quality of the disambiguation results;



Node	Term	Synset
concept	concept	1. #5498459
computers	computer	1. #2971313
desktopPCComponents	desktop	2. #2673077
desktopPCComponents	PC	1. #3770644
desktopPCComponents	component	3. #2969395
memory	memory	4. #3604415
speaker	speaker	2. #3554056
fan	fan	1. #3199305
mouse	mouse	2. #3651402
cameras	camera	1. #2838071
accessories	accessory	1. #2580125
batteries	battery	2. #2711724
antiques	antique	2. #2629849
musical instruments	musical instrument	1. #3658031
string	string	2. #2778327
furniture	furniture	1. #3281059
chairs	chair	1. #2894438

Synset	Long description	Hypernym hierarchy	Score	Sel
2. #3651402	*mouse, computer_mouse — (a hand-operated electronic device that controls the coordinates of a cursor on your computer screen as you move it around on a pad; on the bottom of the mouse is a ball that rolls on the surface of the pad; "a mouse takes much more room than a trackball")*	*-> mouse, computer_mouse -> electronic_device -> device -> instrumentality, instrumentation -> artifact, artefact -> object, physical_object -> entity -> whole, whole_thing, unit -> object, physical_object -> entity*	75 %	☑
1. #2244572	*mouse — (any of numerous small rodents typically resembling diminutive rats having pointed snouts and small ears on elongated bodies with slender usually hairless tails)*	*-> mouse -> rodent, gnawer, gnawing_animal -> placental, placental_mammal, eutherian, eutherian_mammal -> mammal -> vertebrate, craniate -> chordate -> animal, animate_being, beast, brute, creature, fauna -> organism, being -> living_thing, animate_thing -> object, physical_object -> entity*	45 %	☑

Fig. 2. The Graphical User Interface of the STRIDER System.

– user-friendly GUI speeding up the *assisted disambiguation* of trees, providing an easy-to-use layout of the informative components.

Technical details about the implemented techniques for structural disambiguation are available in [4].

3 Demonstration

In this section we demonstrate the main features of STRIDER. The effectiveness of the system has been experimentally measured on several trees differing in the level of specificity and polysemy [4] (trees are available online at `www.isgroup.unimo.it/paper/strider`).

Fig. 2 shows STRIDER's GUI with the results of the disambiguation process for the eBay example (Fig. 1-a). In the left part of the GUI we see columns `Node`, `Term` that show the outcome of the automated extraction of terms from the tree's nodes and column `Synset` that contains the chosen sense for the corresponding term. For flexibility purposes, the GUI allows users to fill it in either by manually choosing one of the senses in the right part or by pressing the *Magic Wand*. This simple act triggers the fully *automatic disambiguation* process of STRIDER which is applied to the entire loaded tree and automatically chooses the top sense in the ranking of each term. When the user highlights a term in the left part of the GUI, the right part shows all the available senses and for each of them the synset's hypernym hyerarchy. 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-a 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". 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. 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% [4]). 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 which is able to deliver a higher disambiguation precision, by expanding the context with additional related nouns contained in the description and in the examples of each sense in WordNet. 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. 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 Conclusions

The disambiguation performances achieved by STRIDER are encouraging and demonstrate the very good effectiveness of the adopted approach. The intuitive GUI provides easy interaction with the user; further, the system can also be used in batch mode to meet the needs of the most cutting edge semantic-aware applications, where user intervention is not feasible.

References

1. P. Cimiano, S. Handschuh, and S. Staab. Towards the self-annotating web. In *Proc. of the 13th WWW Conference*, 2004.
2. H. Do, S. Melnik, and E. Rahm. Comparison of schema matching evaluations. In *Proc. of the 2nd WebDB Workshop*, 2002.
3. Marc Ehrig and Alexander Maedche. Ontology-focused crawling of web documents. In *Proc. of the ACM SAC*, 2003.
4. F. Mandreoli, R. Martoglia, and E. Ronchetti. Versatile Structural Disambiguation for Semantic-aware Applications. In *Proc. of the 14th CIKM Conference*, 2005.
5. F. Mandreoli, R. Martoglia, and P. Tiberio. Approximate Query Answering for a Heterogeneous XML Document Base. In *Proc. of the 5th WISE Conference*, 2004.
6. G. A. Miller. WordNet: A Lexical Database for English. *CACM*, 38(11), 1995.
7. I. Tatarinov and A. Halevy. Efficient Query Reformulation in Peer Data Management Systems. In *Proc. of ACM SIGMOD*, 2004.
8. 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.