Word Sense Disambiguation for XML Structure Feature Generation

Andrea Tagarelli, Mario Longo, Sergio Greco

Dept. of Electronics, Computer and Systems Sciences, University of Calabria, Italy
e-mail: tagarelli@deis.unical.it

Abstract. A common limit of most existing methods that manage XML structure information is that they do not handle the semantic meanings that might be associated to the markup tags. In this paper, we study how to map structure information available from XML elements into semantically related concepts in order to support the generation of XML semantic features of XML structural type. For this purpose, we define an unsupervised *word sense disambiguation* method to select the most appropriate meaning for each element contextually to its respective XML path. The proposed approach exploits conceptual relations provided by a lexical ontology such as WordNet and employs different notions of *sense relatedness*. Experiments with data from various application domains are discussed, showing that our approach can be effectively used to generate structural semantic features.

1 Introduction

Structure information is essential in many problems of semistructured data management, such as indexing, query processing, change detection and schema matching. There is also a variety of tasks of learning and understanding from XML data (e.g., similarity detection, summarization, classification and clustering) focusing on the structural characteristics of such data.

As XML becomes increasingly widespread, handling the semantics of XML data arises as one of the most difficult challenges in data management and knowledge discovery. Indeed, XML data exhibiting proper structures and text contents may in principle encode related semantics due to a subjective definition of markup tags. Within this view, a common limit of most existing approaches is that they represent XML data according to syntactic structural information, whereas they do not consider an important characteristic of XML data: the semantic meanings that might be associated to the markup tags.

In this paper, we address the problem of generating semantic features for XML data focusing on structure information only. A major assumption of our work is that a *lexical ontology*, such as WordNet [4], can play a central role in identifying semantic relationships among the concepts underlying the constituents of structure information. A concept is represented by a lexical meaning (sense) which can be assigned to one or more terms. In the proposed structural analysis, these terms correspond to the tag names in XML paths and the objective is to couple syntactic information (the tag name) with a semantic one (the concept associated to the tag name).

To address the inherent ambiguity of the meaning of tag names, we have devised a *word sense disambiguation* method to select the most appropriate sense for each tag name in the context of an XML path. This method is based on different notions of *sense relatedness*, which can exploit scoring functions for overlaps between dictionary glosses and distance measures for ontology paths.

We have experimentally evaluated our approach on various data belonging to different application domains. Results have raised the significance of our approach and provided important indications on the relative roles of the devised notions and methods.

2 Related work

Representing semistructured data is typically addressed by labeled rooted trees. Consequently, in the past few years, handling such data has leveraged results from research on tree matching, including a number of algorithms for computing tree edit distances (e.g., [9]). More recently, attention has also been drawn toward using simple Vector-space models to represent XML data, which substantially differ in the definition of feature space (e.g., [3, 16, 2]). However, in order to represent the structural characteristics of XML data, all the aforementioned approaches use syntactic information only, while ignoring semantic information.

The key element in many tasks involving a notion of semantics has been identified in *ontologies*. In particular, *WordNet* [4] is widely known as the most important publicly available large-scale lexical ontology. In WordNet, related concepts are grouped into equivalence classes, called *synsets* (sets of synonyms). Each synset represents one underlying lexical concept and is described by a short text (*gloss*). Synsets are explicitly connected to each other in the form of ontologies through different relations, such as *is-a* relations (hypernymy/hyponymy), *part-of* relations (meronymy/holonymy), etc. In this context, an early study is presented in [15], where texts and structure information are combined together to generate XML structure and content features, although ontological knowledge is marginally used. In the XML structural clustering approach proposed in [8], an element name is associated with a set containing its constituent words and semantically related words (e.g., the synonyms, hyponyms and hypernyms for each of the constituent words). Given any two element names, the similarity is computed by considering the intersection between their respective word sets. In the case that there are no common words between such sets, the similarity is computed by averaging the pair-wise syntactic matchings based on a combination of edit distances and n-gram functions.

In our previous works [14, 13], we have addressed the problem of semantic relatedness between XML documents in a transactional clustering framework, focusing on the generation of semantically-enriched structure and content features.

In particular, we originally involved an unsupervised analysis of the meanings of the tag names into the structural feature generation.

Semantic analysis of XML structural characteristics naturally involves the possible senses of the tag names. This may eventually require a task of *word sense disambiguation* (WSD) in order to assign each tag with the most appropriate sense in a given context. In dictionary-based WSD the assumption is that the most plausible sense to assign to multiple co-occurring words is the one that maximizes the relatedness among the chosen senses. Within this view, the pioneer Lesk method [5] disambiguates a target word by choosing the meaning whose gloss shares the largest number of words with the glosses of the neighboring words. However, using a lexical ontology like WordNet allows for capturing semantic relationships based on concepts by also exploiting hierarchies of concepts besides dictionary glosses. The basic Lesk algorithm can be enhanced in this way to take advantage of the network of relations provided in WordNet. This idea has been formalized in a measure of semantic relatedness between word senses based on the notion of *extended gloss overlap* [10, 1], which has the merit of considering phrasal matches and weighting them more heavily than single word matches. The extended gloss overlap measure takes as input two concepts (i.e., WordNet synsets) and computes a gloss overlap score, hereinafter denoted as *go-score*, as the sum of the squared sizes of the distinct overlaps between the glosses, where any *overlap* is detected whenever a shared maximal sequence of words occurs. Other functions of semantic relatedness/similarity for WSD problems have recently been proposed in, e.g., [17, 11].

The study in [7] describes a structural approach to sense classification. A WSD algorithm based on a context-free grammar is developed to find structural semantic interconnections, i.e., structural specifications of the possible senses for each word in a context. The graph representation of word senses is built from several sources, including WordNet but also annotated corpora and glossaries. This makes the disambiguation approach semi-supervised. A recent attempt to the disambiguation of tag names in XML trees has been proposed in [6]. The disambiguation of any given tag is accomplished in an unsupervised way by analyzing the relative structural context with the support of an external knowledge source such as WordNet. The structural context of each word appearing in the target tag is a graph built on the ancestors, descendants and siblings of its corresponding node. Each word in the target tag is compared to all the words derived from this graph-context in order to calculate a vector of similarity scores associated to the possible senses. This vector is seen as a ranking of the plausible senses for each term, which can further be refined by considering the gloss definitions and linear ordering of the noun synsets.

Like [6], our work proposes a method of unsupervised disambiguation of XML elements, which uses WordNet as ontological knowledge. However, the use of WordNet in [6] is limited to synonymies and is-a relations, whereas our method involves other concept hierarchies (e.g., part-of relations). Also, we devise different notions of sense relatedness and various strategies of search through the WordNet hierarchies.

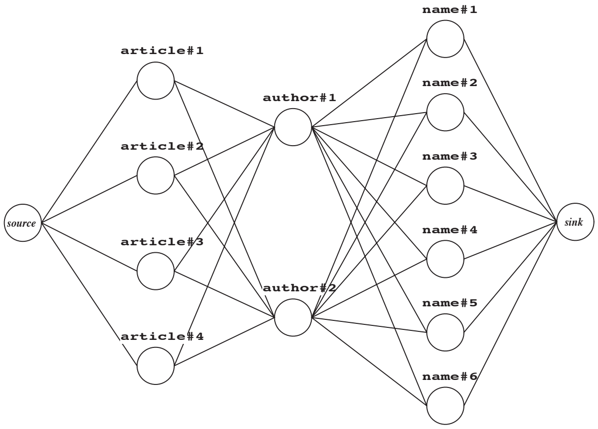


Fig. 1. The synset graph for the path `article.author.name`. (Note that edge weights are not shown to avoid cluttering for clear presentation)

3 Tag sense disambiguation

Tag paths represent the natural basis for extracting structure features from XML data. Since we desire to go beyond a context-free use of syntactic terms (tag names), we aim to map them into semantically related concepts.

For any given tag path p, the objective is to provide each tag in p with a unique sense chosen from the reference lexical ontology; we assume that a fictitious sense is assigned to any term (tag name) for which no matches are found in the reference lexical ontology. Since selected senses are to be appropriate contextually to p, we need a task of WSD to handle the various senses of any given tag name and to finally select the most appropriate one with respect to a *path context*.

3.1 Building the synset graph

We conceive a path context as a "semantic network" which is built on the senses of the tags of a given path. More precisely, this network is modeled as a weighted graph of layered form, called *synset graph*, such that: *i)* each layer consists of all the possible senses of any given tag in the path, *ii)* the layers are connected according to the order of the tags in the path, and *iii)* the edge weights are computed in such a way that they measure the relatedness between nodes (synsets) belonging to adjacent layers. Figure 1 shows an example synset graph.

3.2 Computing the edge weights in a synset graph

The crucial aspect in building a synset graph is how to compute the edge weights. For this purpose, we pursue two ideas.

The first idea is to exploit the mechanism of *gloss overlap scoring* (i.e., the *go-score* computation discussed in Section 2) which is possibly enhanced by using the WordNet concept hierarchies.

The second idea is to define a notion of sense relatedness which is evaluated with respect to WordNet by applying an *ontology-path-based similarity* measure (e.g., [12]). The degree to which two senses are related is seen as a function of their respective locations in the lexical ontology, in such a way that the higher a sense in a hierarchy the more general it is.

Gloss-overlap-based sense relatedness. Let $\mathcal{WSR}$ denote a selected set of WordNet synset relations, namely *hypernymy*, *hyponymy*, *meronymy*, and *holonymy*. Given a relation $r \in \mathcal{WSR}$ and a synset σ, we define a function ω such that, when applied to r and σ, it yields the set $\omega(r, \sigma)$ of synsets directly connected to σ through r.

Function ω can be enhanced to include synsets that are indirectly connected to a target synset to a given distance in the WordNet taxonomy. Given a relation $r \in \mathcal{WSR}$, a synset σ and an integer $d \geq 1$, the set of synsets σ' connected to σ, through r, across a path of length d is defined as $\omega^*(r, \sigma, d) = \bigcup_{\sigma' \in \omega(r,\sigma)} \omega^*(r, \sigma', d-1)$, if $d > 1$; otherwise, $\omega^*(r, \sigma, d) = \omega(r, \sigma)$.

Given any two synsets σ and ρ, a non-negative integer $\overline{d}$, and a function $\mathrm{f}(d)$ monotonically decreasing for increasing values of d, we define the gloss-overlap-based sense relatedness as:

$$
\begin{aligned}
&\textit{go-relatedness}(\sigma, \rho, \overline{d}) = \textit{go-score}(\sigma, \rho) \\
&+ \sum_{d=1}^{\overline{d}} \Bigg[\sum_{\substack{\rho' \,\in\, \omega^*(\textit{hypernymy}, \rho, d) \,\vee \\ \rho' \,\in\, \omega^*(\textit{holonymy}, \rho, d)}} \textit{go-score}(\sigma, \rho') \times \mathrm{f}(d) \Bigg] \\
&+ \sum_{d=1}^{\overline{d}} \Bigg[\sum_{\substack{\sigma' \,\in\, \omega^*(\textit{hyponymy}, \sigma, d) \,\vee \\ \sigma' \,\in\, \omega^*(\textit{meronymy}, \sigma, d)}} \textit{go-score}(\sigma', \rho) \times \mathrm{f}(d) \Bigg]
\end{aligned}
$$

A simple enhancement to the above measure can be made by introducing a variant in the *go-score* computation: for any synset σ, the set of synonyms associated with σ can be included in the set of terms defining the σ's gloss. In this way, a gloss can be enriched with more, semantically equivalent terms.

Ontology-path-based sense relatedness. We believe that searching through the (WordNet) is-a hierarchy upwardly and/or through the part-of hierarchy is useful to capture relatedness among synsets.

Given any two synsets σ and ρ, we define the ontology-path-based relatedness between σ and ρ as:

$$
\textit{p-relatedness}(\sigma, \rho) = \max_r \left\{ \frac{2 \times \textit{depth}_r(\textit{lcs}_r(\sigma, \rho))}{\textit{depth}_r(\sigma) + \textit{depth}_r(\rho) + |\textit{lcs}_r(\sigma, \rho)| - 1} \right\}
$$

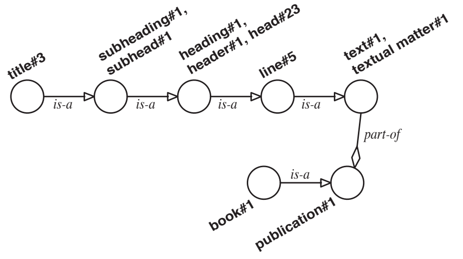


Fig. 2. The portion of WordNet explored to compute the ontology-path-based relatedness between the 1th synset of `book` and the 3rd synset of `title`

where

- $r \in \{hypernymy, holonymy\}$.
- $depth_r()$ computes the maximum distance (path length) from a given synset to a top hypernym (if $r = hypernymy$) or top holonym (if $r = holonymy$). The depth of a top synset is set to 1, whereas the depth of a synset whose sense is not found in the hierarchy is assumed to be 0.
- $lcs_r()$ computes the set of lowest common synsets between two given synsets with respect to r. This set contains a unique synset if a common hypernym ($r = hypernymy$) or holonym ($r = holonymy$) exists. Otherwise, if $r = hypernymy$, the nearest hypernym of σ and the nearest hypernym of ρ are identified in such a way that they are connected through a minimum-length chain of holonyms/meronyms; the *lcs* hence contains these hypernyms and the related holonyms/meronyms. Analogously, if $r = holonymy$, the *lcs* contains the nearest holonyms and the related hypernyms/hyponyms.

It easy to see that the *p-relatedness* function is defined to capture not only relationships involving the is-a and part-of hierarchies distinctly. Indeed, semantic relatedness between senses may be found through both the is-a and part-of hierarchies.

Figure 2 shows the portion of WordNet which is explored to compute the semantic relatedness between the synsets of `book` and `title`. Note that no is-a or part-of direct relationship holds for the two target synsets. Therefore, part-of relationships are to be searched among `book`#1's hypernyms and `title`#3's hypernyms. In this way, a 4-level hypernym of `title`#3, that is `text`#1 `textual matter`#1, is found as a meronym of a hypernym of `book`#1, that is `publication`#1. Thus, `text`#1 `textual matter`#1 and `publication`#1 represent the *lcs* for the two target synsets.

Definitions of synset graph edge weight. Let $a = \langle t_i, \sigma \rangle$ and $b = \langle t_{i+1}, \rho \rangle$ be synsets corresponding to connected nodes in a synset graph. The weight on the edge (a, b) can be defined in various ways according to the different settings of the two approaches previously discussed. In the following, for each definition we fix the corresponding notation within brackets, which will be used in the presentation of the experiments.

- Gloss-overlap-based definitions — various forms of the function $\mathrm{f}(d)$ can be chosen, such as exponential functions. Also, the parameter $\overline{d}$ will be tuned experimentally. We provide here the following definitions:
  - [G] $weight(a, b) = \textit{go-relatedness}(\sigma, \rho, 0) \equiv \textit{go-score}(\sigma, \rho)$.
  - [G_Exp] $weight(a, b) = \textit{go-relatedness}(\sigma, \rho, \overline{d})$, with $\mathrm{f}(d) = e^{-d}$.
  - [G_SqExp] $weight(a, b) = \textit{go-relatedness}(\sigma, \rho, \overline{d})$, with $\mathrm{f}(d) = e^{-d^2}$.
  - [G_InvLin] $weight(a, b) = \textit{go-relatedness}(\sigma, \rho, \overline{d})$, with $\mathrm{f}(d) = 1/(1 + d)$.

  Moreover, for each of the definitions above, we need to decide whether a gloss is to be enhanced with the corresponding set of synonyms; if this is the case, we will use the notation suffix /+S.
- Ontology-path-based definition — this setting corresponds to the definition [P] $weight(a, b) = \textit{p-relatedness}(\sigma, \rho)$.
- Composite definitions — the *go-relatedness* and *p-relatedness* measures can be combined together in various ways. We provide here the following definitions:
  - [P_cnd] $weight(a, b) = \textit{go-relatedness}(\sigma, \rho)$, if $\textit{p-relatedness}(\sigma, \rho) > 0$; otherwise, $weight(a, b) = 0$. Here we are assuming that the sense relatedness is computed on the basis of the gloss overlap method conditionally to the path-based relatedness.
  - [GxP] $weight(a, b) = \textit{go-relatedness}(\sigma, \rho) \times \textit{p-relatedness}(\sigma, \rho)$. In this function the path-based term acts as a damping factor for the gloss-based term, since *p-relatedness* assumes values in [0..1].

Disambiguating the path tags. Once the synset graph for a given XML path p has been established, the disambiguation of all the tag names in p is accomplished by finding the "best" path in the graph. The senses corresponding to this graph path are recognized as the most appropriate senses for the tag names in p. We devise two ways to find the "best" path in a synset graph:

- *Maximum-weight path* [MWP]: the best path is such that the sum of the weights over its edges is maximum. In the case of multiple best paths in the synset graph, the preferred path can be computed by exploiting the dictionary-supplied linear order of synsets associated with the tag names.
- *Direction-driven path*: the best path is detected as the one resulting from a search of the maximum weight layer-by-layer in the graph; precisely, once the maximum weight has been found in a certain edge of the first layer visited, this edge fixes a portion of the best path being detected, and the search analogously continues on the next layer. Clearly, this search depends on the direction the path is visited, namely "from left" [DDP_sx] (i.e., top-down in the XML tree) or "from right" [DDP_dx] (i.e., bottom-up in the XML tree).

The maximum-weight method appears to be more general than the direction-driven method; however, the latter could in principle perform better in some cases, since it takes into account the specific order of the tags in an XML path, either in a top-down or bottom-up fashion.

Note also that in the case of weights equal to zero for all the edges between any two adjacent tags, a maximum weight is set for the edge between the first senses of the given tags—which is in agreement with the dictionary-supplied linear order of the synsets.

4 Experimental evaluation

We collected various XML data belonging to different application domains. In this section, we present experimental results obtained on XML structures whose schemas are shown in Fig. 3:

- *DBLP*: data concerning scientific bibliography with a DBLP-like structure.[1]
- *Reuters*: news headlines from the Reuters RSS news channel.[2]
- *People*: customized data structure for recording personal information.
- *Wikipedia*: data representing encyclopedia articles with a Wikipedia-like structure.
- *Shakespeare*: plays from the Shakespeare 2.00 collection.[3]

In principle, tag names should be subject to a number of text processing operations. Removal of stopwords might be performed, as long as such stopwords do not represent lexical constituents of a compound term; as an example, given the tag `state-of-the-art`, the stopwords "of" and "the" should not be removed from that tag. Word splitting can also be useful; in this work, we assume to perform a semi-automatic operation of word splitting, which is driven by both the tag tokenization based on delimiters (e.g., hyphen, underscore, or variation in the letter case) and ad-hoc lists of domain-specific compound terms.

We evaluated the results of tag sense disambiguation by comparing them with a human disambiguation based on the latest version of WordNet. More precisely, for each dataset, we manually selected the appropriate senses for the tag terms according to our comprehension of the meanings of the synset glosses. These senses were used as reference for the evaluation, as shown in Fig. 4. It should be noted that we selected more than one "ideal" sense for some tag terms in a specific data. This was mainly due to our difficulty in capturing the underlying relationships between the tag terms which may depend on a subjective interpretation of the synset descriptions.

To assess the accuracy in the disambiguation, we define a measure that considers the particular form of the disambiguation context (i.e., the synset graph). This measure, called *tag-pair-accuracy*, computes for each path the fraction of pairs of adjacent tags in the path which have been correctly disambiguated.

[1] http://www.informatik.uni-trier.de/∼ley/db/

[2] http://www.reuters.com/tools/rss

[3] http://metalab.unc.edu/bosak/xml/eg/shaks200.zip

```
<!ELEMENT bibliography (article | inproceedings | book)+>
<!ELEMENT article (author+, title, journal, volume, year, publisher)>
<!ELEMENT inproceedings (author+, title, pages, year, booktitle)>
<!ELEMENT book (author+, title, publisher, year, editor?, series?, volume?)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT booktitle (#PCDATA)>
<!ELEMENT editor (#PCDATA)>
<!ELEMENT journal (#PCDATA)>
<!ELEMENT pages (#PCDATA)>
<!ELEMENT publisher (#PCDATA)>
<!ELEMENT series (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT volume (#PCDATA)>
<!ELEMENT year (#PCDATA)>
```

(a) *DBLP*

```
<!ELEMENT news (channel+)>
<!ELEMENT channel (title, link?, description, image?, item+)>
<!ELEMENT image (title, width, height, link, url)>
<!ELEMENT item (title, description)>
<!ELEMENT description (#PCDATA)>
<!ELEMENT height (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT url (#PCDATA)>
<!ELEMENT width (#PCDATA)>
```

(b) *Reuters RSS*

```
<!ELEMENT Shakespeare (play+)>
<!ELEMENT play (title, prologue?, act*)>
<!ELEMENT act (scene+, epilogue?)>
<!ELEMENT prologue (title, speech)+>
<!ELEMENT epilogue (title, speech)+>
<!ELEMENT scene (title, speech)+>
<!ELEMENT speaker (#PCDATA)>
<!ELEMENT speech (speaker)>
<!ELEMENT title (#PCDATA)>
```

(c) *Shakespeare*

```
<!ELEMENT encyclopedia (article+)>
<!ELEMENT article (header, body)>
<!ELEMENT header (title, revision, categories)>
<!ELEMENT body (section+)>
<!ELEMENT section (title, link?, list*, figure*)>
<!ELEMENT categories (category+)>
<!ELEMENT figure (image, caption)>
<!ELEMENT caption (#PCDATA)>
<!ELEMENT category (#PCDATA)>
<!ELEMENT entry (#PCDATA)>
<!ELEMENT image (#PCDATA)>
<!ELEMENT link (#PCDATA)>
<!ELEMENT list (entry+)>
<!ELEMENT revision (#PCDATA)>
<!ELEMENT title (#PCDATA)>
```

(d) *Wikipedia*

```
<!ELEMENT people (person+)>
<!ELEMENT person (name, address, email, phonenumber, creditcard?, profile)>
<!ELEMENT address (street, city, country, province, zipcode)>
<!ELEMENT profile (gender, age, income, education, business)>
<!ELEMENT age (#PCDATA)>
<!ELEMENT business (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT creditcard (#PCDATA)>
<!ELEMENT education (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT gender (#PCDATA)>
<!ELEMENT income (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ATTLIST person id NMTOKEN #REQUIRED>
<!ELEMENT phonenumber (#PCDATA)>
<!ELEMENT province (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT zipcode (#PCDATA)>
```

(e) *People*

Fig. 3. DTDs associated with the XML data used in the experiments

tag	*data*	*senses*
act	*Shakespeare*	3
address	*People*	2, 6
age	*People*	3
article	*DBLP*	1
article	*Wikipedia*	1, 3
author	*DBLP*	1
bibliography	*DBLP*	1
body	*Wikipedia*	9
book	*DBLP*	1
business	*People*	6
caption	*Wikipedia*	3
categories	*Wikipedia*	1, 2
category	*Wikipedia*	1, 2
channel	*Reuters*	5, 7, 8
city	*People*	1
country	*People*	1
credit_card	*People*	1
description	*Reuters*	1, 3
editor	*DBLP*	1
education	*People*	1, 3
email	*People*	1
encyclopedia	*Wikipedia*	1
entry	*Wikipedia*	1
epilogue	*Shakespeare*	1, 2
figure	*Wikipedia*	1
gender	*People*	2
header	*Wikipedia*	1
height	*Reuters*	1, 3
id	*People*	2
image	*Reuters*	2
image	*Wikipedia*	2
income	*People*	1
item	*Reuters*	1, 4
journal	*DBLP*	2
link	*Reuters*	1, 7
link	*Wikipedia*	6, 7
list	*Wikipedia*	1
name	*People*	1
news	*Reuters*	3, 4
pages	*DBLP*	1
people	*People*	1
person	*People*	1
phone_number	*People*	1
play	*Shakespeare*	1
proceedings	*DBLP*	2
profile	*People*	3
prologue	*Shakespeare*	1
province	*People*	1
publisher	*DBLP*	2
revision	*Wikipedia*	3
scene	*Shakespeare*	6, 1
section	*Wikipedia*	1
series	*DBLP*	3
Shakespeare	*Shakespeare*	1
speaker	*Shakespeare*	1
speech	*Shakespeare*	7, 1
street	*People*	1, 2
title	*DBLP*	3
title	*Reuters*	3, 1
title	*Shakespeare*	3, 2
title	*Wikipedia*	3
url	*Reuters*	1
volume	*DBLP*	4, 3
width	*Reuters*	1
year	*DBLP*	1
zip_code	*People*	1

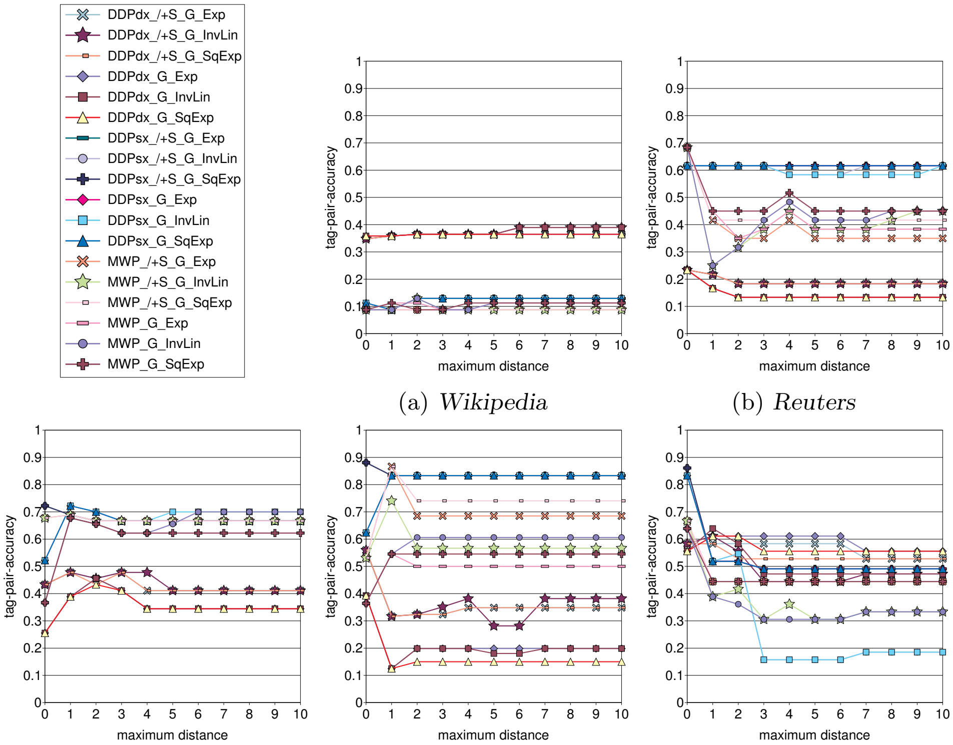
Fig. 4. The senses selected for the evaluation

The overall tag-pair-accuracy is finally obtained by averaging the local accuracy values over all the distinct paths in an XML document.

We would like to point out that we have chosen a single assessment criterion in this work for the sake of brevity. However, the notion of tag-pair-accuracy has been preferred to other, more standard criteria such as the classic precision, which evaluate the contingencies of single instances (i.e., the tag names) rather than pairs of instances. This results in an assessment criterion which is in principle tougher than precision or other similar measures, thus causing a relative underestimation of the disambiguation results.

4.1 Evaluation of gloss-overlap-based disambiguation

Figure 5 shows the accuracy results obtained by using the gloss-overlap-based disambiguation. In general, the form of the $\mathrm{f}(d)$ function did not affect the accuracy substantially. The option /+S turned out to be beneficial to the disambiguation in nearly all the datasets. Regardless of $\mathrm{f}(d)$, $\overline{d}$ and the strategy of



(a) *Wikipedia*

(b) *Reuters*

(c) *People*

(d) *Shakespeare*

(e) *DBLP*

Fig. 5. Accuracy results using gloss-overlap-based methods

search in the synset-graph, the average improvement was up to 11% on *Shakespeare*, 6% on *People*, 0.7% on *Reuters*, 0.6% on *DBLP*. However, the accuracy improvement was even higher when the best search strategy was used (e.g., 17% on *Shakespeare* for DDP_dx, 8% on *People* for DDP_dx, 4% on *Reuters* for DDP_dx). For example, the expected disambiguation for `speech` and `speaker` (i.e., synset #7 and #1, respectively) was captured thanks to the enhancement of the `speech`#7's gloss description, which leads to the overlap 'speech' with the `speaker`#1's gloss description.

The accuracy trends mainly depended on the value of the maximum distance $\overline{d}$ and the strategy of search in the synset-graph. In general, the accuracy tended to stabilize as $\overline{d}$ increases, especially for $\overline{d} \geq 2$. However, in some cases, the accuracy may decrease as $\overline{d}$ increases depending on the search strategy. For example, on *DBLP*, regardless of f(d) and /+S the difference between the maximum and the minimum accuracy was around 8% for DDP_dx, 14% for MWP and 31% for DDP_sx. Considering an average over f(d) and /+S, the best accuracy was

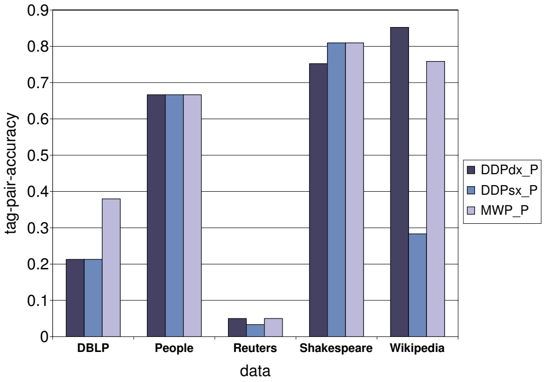


Fig. 6. Accuracy results using ontology-path-based methods

achieved for small $\overline{d}$, in particular: 0.85 on *DBLP* for $\overline{d} = 0$, 0.88 on *Shakespeare* for $\overline{d} = 1$, 0.74 on *People* for $\overline{d} = 1$, 0.68 on *Reuters* for $\overline{d} = 0$.

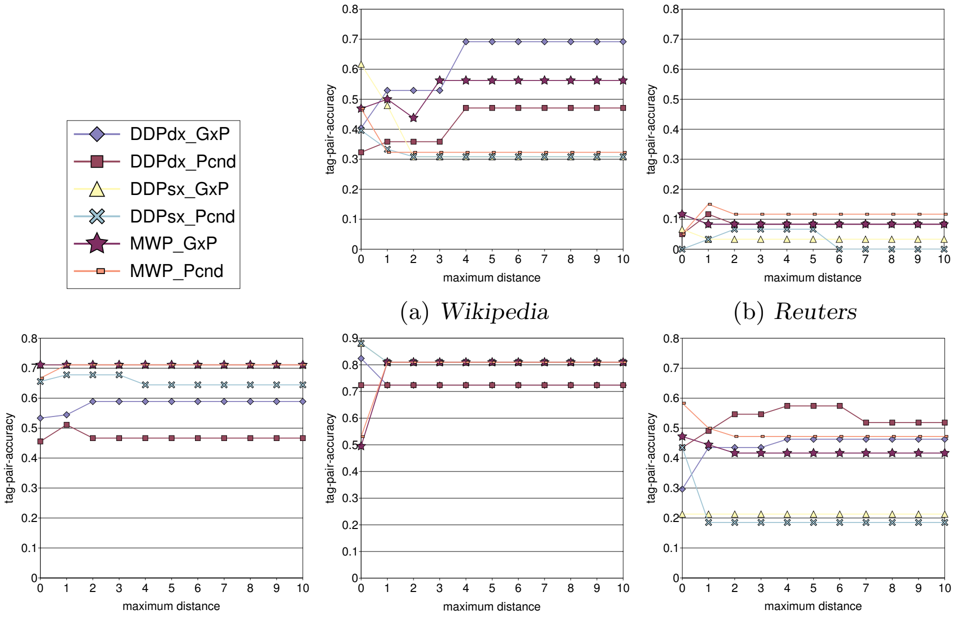
As concerns the strategy of search in the synset-graph, DDP_sx behaved better than DDP_dx and MWP in all the datasets (except *Wikipedia*) considering the accuracy values averaged over $\overline{d}$, f(d) and /+S. In particular, comparing DDP_sx to DDP_dx, the improvement was up to 51% on *Shakespeare*, 44% on *Reuters*, 30% on *People*, 11% on *DBLP*. However, it should be noted that DDP_dx was less sensitive to $\overline{d}$, f(d) and /+S than DDP_sx and MWP.

A special remark has to be made on *Wikipedia*. In this data we observed a failure of the gloss-overlap-based disambiguation due to the lack of "significant" overlaps, that is overlaps consisting of false content-bearing words. For example, synsets `encyclopedia`#1 and `article`#4 share the term 'reference', and `article`#4 and `title`#1 share the term 'may', which mislead to disambiguate the meanings of such tags in their context. The accuracy was further negatively affected (especially when MWP and DDP_sx were used) by the fact that `encyclopedia`, `article` and `title` form a common prefix of XML paths in *Wikipedia*.

4.2 Evaluation of ontology-path-based disambiguation

The ontology-path-based disambiguation may take advantage of some characteristics of certain data. Comparing the results in Fig. 6 with Fig. 5, significant improvements were observed in some cases, in particular: on *Wikipedia*, up to 60% for MWP, 50% for DDP_dx and 10% for DDP_sx; on *Shakespeare*, up to 30% for DDP_dx and 15% for MWP; on *People* up to 20% for DDP_dx. By contrast, the accuracy drastically decreased on *Reuters* and *DBLP* regardless of the search strategy used.

We tend to justify these different behaviors by observing the cohesiveness of the path tags with respect to a specific application domain. Indeed, most tags in *Wikipedia* (e.g., `encyclopedia.article`), *Shakespeare* (e.g., `act.scene.speech`)



(a) *Wikipedia* (b) *Reuters*

(c) *People* (d) *Shakespeare* (e) *DBLP*

Fig. 7. Accuracy results combining gloss-overlap- and ontology-path-based methods

and, partly, *People* (e.g., `zipcode`, `address`, `phonenumber`) show high pertinence to their respective domains. By contrast, some tag names in *Reuters* typically occur in different contexts with multiple meanings; in particular, when these tag names are adjacent and form prefixes of path (e.g., `channel` and `item`), the accuracy of disambiguation will result very low.

4.3 Evaluation of combined relatedness

Figure 7 shows the results obtained by combining weighting methods of the approaches previously discussed—we present here results referring to the case $f(d) = e^{-d}$ and gloss enhancement (option /+S). In general, using *p-relatedness* was relevant for the disambiguation, since the accuracy achieved average values similar to those observed with the pure ontology-path-based approach.

Compared to the results obtained by using gloss-overlap-based methods, the accuracy significantly improved in *Wikipedia* (from 40% to 70%), while substantially did not vary in *Shakespeare* and *People*. A drastic decrease in accuracy occurred in *Reuters* (which was due to the lack of *p-relatedness* as previously observed in Fig. 6), while the performance of *DBLP* tended to decrease depending on the search strategy and the composite methods.

Focusing the attention on the composite methods, we observed there was no particular setting in which one method prevailed against the other one, averaging over $\overline{d}$. For instance, on *DBLP*, P_cnd led to better average results when DDP_dx or MWP was used, whereas GxP slightly prevailed for DDP_sx and higher $\overline{d}$. On *Reuters*, P_cnd performed as good as or better than GxP. On *Wikipedia* and *People*, GxP performed as good as or better than P_cnd; in particular, on *Wikipedia*, a good value of *p-relatedness* lowered the impact of "false" overlaps, leading to relatively high performance for DDP_dx and GxP. In general, P_cnd seemed to be less sensitive to $\overline{d}$ than GxP.

4.4 Lessons learned

Facing with the above results, we can summarize the following main remarks:

- Using the set of synonyms to enhance the text of a gloss definition is effective since it increases the probability of finding (semantically useful) overlaps between glosses.
- The maximum distance by which a synset is compared with relating indirect synsets may affect the disambiguation accuracy in different ways depending on the conceptual relatedness (*p-relatedness*) among the path tags; however, in general, accuracy tends to stabilize with $\overline{d} \geq 3$, and may have maximum peaks with smaller $\overline{d}$ in some cases.
- The form of the damping function $\mathrm{f}(d)$ is not crucial to the disambiguation, which is probably due to a small contribution of the gloss overlaps using indirect synsets.
- Different choices of the strategy of search in the synset graph may result in different disambiguations. DDP_sx and MWP tend to lead to better performance, whereas DDP_dx seems to be more robust to variations that may occur in the other parameters.

It is worth emphasizing that measuring the tag sense relatedness by using the gloss-overlap-based methods has proved to be effective, although it may lead to scarcely or not significant scorings; indeed, gloss descriptions may be not enough to capture relatedness between close synsets, and may also contain false content-bearing words. On the other hand, using only the WordNet concept hierarchies may have a very different impact on the disambiguation accuracy: in general, the higher the domain homogeneity of the path tags, the more significant the contribution of the WordNet hierarchies (especially the *is-a* hierarchy) to achieve better disambiguation.

5 Conclusion

We investigated how to semantically-enrich structure information available from XML data in order to support the generation of XML semantic features. In our approach, lexical ontology (WordNet) plays a central role in identifying semantic relationships among the concepts underlying the constituents of structure information. A method of word sense disambiguation is defined to couple each tag name with its appropriate meaning in the context of an XML path. We presented an experimental evaluation of the proposed approach and discussed some lessons learned.

We plan to extend our approach in several directions. One possible concerns a simultaneous analysis of all the distinct paths in an XML document, which could aid to adaptively disambiguate the tag senses with respect to the context of the document. We also intend to enhance the notions of sense relatedness with information on the tag-concept domains available from WordNet or other ontologies.

References

1. S. Banerjee and T. Pedersen. Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proc. IJCAI*, pages 805–810, 2003.
2. L. Candillier, I. Tellier, and F. Torre. Transforming XML Trees for Efficient Classification and Clustering. In *INEX Workshop*, pages 469–480, 2005.
3. A. Doucet and M. Lehtonen. Unsupervised Classification of Text-Centric XML Document Collections. In *INEX Workshop*, 2006.
4. C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.
5. M. Lesk. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to tell a pine cone from a ice cream cone. In *Proc. ACM SIGDOC Int. Conf. on Systems Documentation*, pages 24–26, 1986.
6. F. Mandreoli, R. Martoglia, and E. Ronchetti. Versatile Structural Disambiguation for Semantic-aware Applications. In *Proc. CIKM*, 2005.
7. R. Navigli and P. Velardi. Structural Semantic Interconnections: A Knowledge-Based Approach to Word Sense Disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1075–1086, 2005.
8. R. Nayak. Fast and effective clustering of XML data using structural information. *Knowledge and Information Systems*, 14:197–215, 2008.
9. A. Nierman and H. V. Jagadish. Evaluating Structural Similarity in XML Documents. In *ACM SIGMOD WebDB Workshop*, pages 61–66, 2002.
10. S. Patwardhan, S. Banerjee, and T. Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. In *Proc. Int. Conf. on Intelligent Text Processing and Computational Linguistics (CICLing)*, pages 241–257, 2003.
11. T. Pedersen, S. Banerjee, and S. Patwardhan. Maximizing Semantic Relatedness to Perform Word Sense Disambiguation. *Tech. rep. UMSI 2005/25, Supercomputing Institute Research at University of Minnesota*, pages 7–9, 2005.
12. P. Resnik. Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11:95–130, 1999.
13. A. Tagarelli and S. Greco. Clustering Transactional XML Data with Semantically-Enriched Content and Structural Features. In *Proc. WISE*, pages 266–278, 2004.
14. A. Tagarelli and S. Greco. Toward Semantic XML Clustering. In *Proc. SIAM Data Mining*, pages 188–199, 2006.
15. M. Theobald, R. Schenkel, and G. Weikum. Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data. In *ACM SIGMOD WebDB Workshop*, pages 1–6, 2003.
16. A. M. Vercoustre, M. Fegas, S. Gul, and Y. Lechevallier. A Flexible Structured-based Representation for XML Document Mining. In *INEX Workshop*, pages 443–457, 2005.
17. D. Yang and D. M. W. Powers. Measuring semantic similarity in the taxonomy of WordNet. In *Proc. Australasian Conf. on Computer Science*, pages 315–322, 2005.