# XSDF: A System for XML Semantic Disambiguation

Nathalie Charbel, Joe Tekli, Richard Chbeir, Gilbert Tekli

# *XSDF*: A System for XML Semantic Disambiguation


Nathalie Charbel
*LIUPPA, Univ. of Pau & Adour Countries, 64600 Anglet, France*
nathalie.charbel@univ-pau.fr

Joe Tekli
*SOE, Lebanese American Univ., 36 Byblos, Lebanon*
joe.tekli@lau.edu.lb

Richard Chbeir
*LIUPPA, Univ. of Pau & Adour Countries, 64600 Anglet, France*
richard.chbeir@univ-pau.fr

Gilbert Tekli
*NOBATEK R&D,*
64600 Anglet, France
gtekli@nobatek.com



***Abstract*—This paper briefly describes and evaluates *XSDF*, a new XML Semantic Disambiguation Framework, taking as input: an XML document and a general purpose semantic network, and then producing as output a semantically augmented XML tree made of unambiguous semantic concepts. Experiments demonstrate the effectiveness of XSDF in comparison with alternative methods.**



***Keywords*—*XML semantic-aware processing; ambiguity degree; sphere neighborhood; semantic network; semantic disambiguation.***


## I. INTRODUCTION

XML-based data processing has been at the center stage of Web-based information retrieval (IR) and database (DB) applications for the past two decades, taking advantage of the semi-structured nature of XML documents to improve the quality of IR and DB data management solutions [9]. Recently, various studies have highlighted the impact of integrating semantic features in XML-based applications, to allow semantic-aware query rewriting and expansion [4, 12], XML document classification and clustering [13, 18], XML schema matching and integration [5, 19], and more recently XML-based semantic blog analysis and event detection in social networks and tweets [1, 2]. Here, a major challenge remains unresolved: *XML semantic disambiguation*, i.e., how to resolve the semantic ambiguities and identify the meanings of terms in XML documents [8]. The problem is made harder with the volume and diversity of XML data on the Web.

Usually, heterogeneous XML data sources exhibit different ways to annotate similar (or identical) data, where the same real world entity could be described in XML using different structures and/or tagging, depending on the data source at hand. For instance (according to a general purpose knowledge base, such as WordNet [6]) the XML tag name *"director"* in document 2 (Fig. 1) can have several meanings, e.g., "*manager of a company"*, *"film director"*, *"theater director"* or *"music director"*. The same goes for most terms/tag names in XML documents 1 and 2, e.g., "*Transcendence*", *"actor"*, *"plot"*, *"cast"*, *"star"*, etc., which can have more than 2 or 3 different semantic senses each, (following WordNet). In essence, the core problem is lexical ambiguity: a term (e.g., an XML element/attribute tag name or data value) may have multiple meanings (homonymy), or may be implied by other related terms (metonymy). Also, several terms can have the same meaning (synonymy) [8].

```
<?xml version= "1.0"?>
<films>
  <picture id= "301">
    <title> Transcendence </title>
    <producer> C. Nolan </producer>
    <year> 2014 </year>
    <genre> Sci-Fi </genre>
    <cast>
      <star> J. Depp </star>
      <star> R. Hall </star>
    </cast>
    <plot> A scientist's drive for AI,
           takes on dangerous
           implications …</plot>
    …
  </picture>
</films>
```

**Doc 1**

```
<?xml version= "1.0"?>
<movies>
  <movie year= "2014">
    <name> Transcendence </Name>
    <director > W. Pfister </Director>
    <produced_By> C. Nolan </Produced_By>
    <actors>
      <actor>
        <firstName>Johnny</FirstName>
        <lastName>Depp</LastName>
      </actor>
      <actor>
        <firstName>Rebecca</FirstName>
        <lastName>Hall</LastName>
      </actor>
    </actors>    …
  </movie>
</movies>
```

**Doc 2**

Fig. 1. Sample documents with different structures and tagging, yet describing the same information.

In this context, *word sense disambiguation* (WSD), i.e., the computational identification of the meaning of words in *context* [11], is central to automatically resolve the semantic ambiguities and identify the intended meanings of XML element/attribute tag names and data values, in order to effectively process XML documents. While WSD has been extensively studied for flat textual data [7, 11], nonetheless, the disambiguation of structured XML data remains largely untouched. The few existing approaches dedicated to XML semantic-aware analysis, e.g., [10, 13-16, 20], have been directly extended from traditional flat text WSD, while maintaining several limitations, motivating this work: i) completely ignoring the problem of *semantic ambiguity*: most existing approaches perform semantic disambiguation on all XML document nodes (which is time consuming and sometimes needless) rather than only processing those nodes which are most ambiguous, e.g., [10, 13-16, 20].; ii) only partially considering the structural relations/context of XML nodes (e.g., solely focusing on parent-node relations [16], or ancestor-descendent relations [14]). For instance, in Fig. 1, processing XML node *"cast"* for disambiguation: considering (exclusively) its parent node label (i.e., *"picture"*), its root node path labels (i.e., *"films"* and *"picture"*), or its node sub-tree labels (i.e., *"star"*), remains insufficient for effective disambiguation; iii) making use of

syntactic processing techniques such as the *bag-of-words* paradigm [13, 16] (commonly used with flat text) in representing XML data as a plain set of words/nodes, thus neglecting XML structural and/or semantic features as well as compound node labels; and iv) existing methods are mostly static in adopting a fixed context size (e.g., parent node [16], or root path [14]) or using preselected semantic similarity measures (e.g., edge-based measure [10], or gloss-based measure [14]), such that user involvement/system adaptability is minimal.

This paper describes *XSDF*, a novel XML Semantic Disambiguation Framework to semantically augment XML documents using a machine-readable semantic network (e.g., WordNet [6], Roget's thesaurus [21], FOAF [1], etc.), identifying the semantic definitions and relationships among concepts in the underlying XML structure, while overcoming the limitations mentioned above. *XSDF* takes as input traditional *syntactic XML trees* and transforms them into *semantic XML trees* (or graphs, when hyperlinks come to play), i.e., XML trees made of concept nodes with explicit semantic meanings. Each concept will represent a unique lexical sense, assigned to one or more XML element/attribute labels and/or data values in the XML document, following the latter's structural context.

The groundwork results and overall design of *XSDF* has been described in [3], and an extended study has been submitted for publication in [17]. In this short paper, we briefly describe *XSDF*'s overall architecture in Section 0 and discuss some experimental results in Section 0.

## II. OVERALL SYSTEM ARCHITECTURE

*XSDF* is as an unsupervised and knowledge-based solution to resolve semantic ambiguities in XML documents. Its overall architecture is shown in Fig. 2 and consists of four modules:

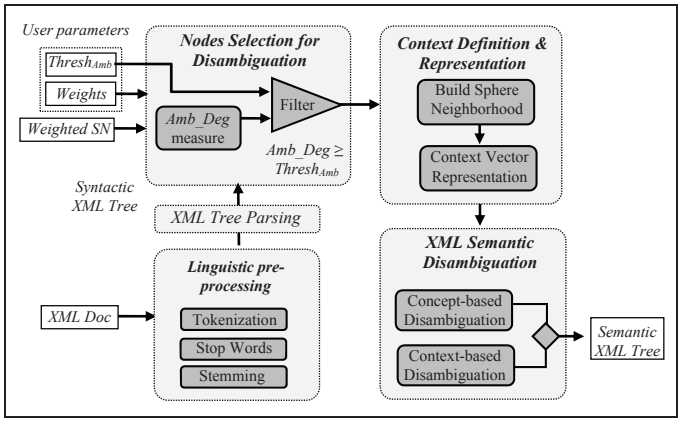
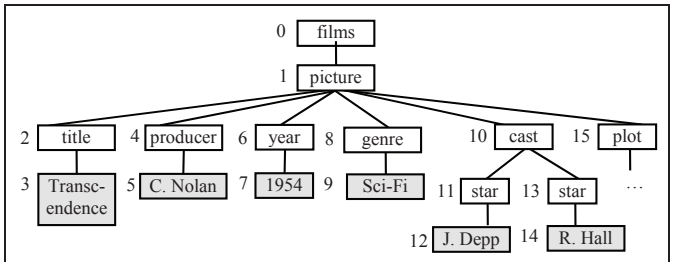



Fig. 2. Overall *XSDF* architecture.

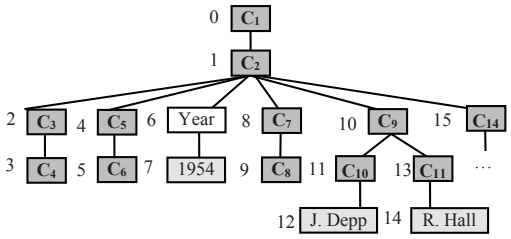
- **Module 1**: *Linguistic pre-processing* of XML node labels and values, performing tokenization, stop words removal, and stemming, as well as handling compound element tag names such that one sense is associated to the tag name, which is formed by the best combination of its compound terms' senses; in contrast with existing studies in [29, 56] which process token senses separately as distinct labels.

- **Module 2**: *Selecting ambiguous XML nodes* as primary targets for disambiguation using a dedicated *ambiguity degree* measure (see in Formula 1) taking into account various structural and semantic factors such as label *polysemy* (the number of senses of the node's label), node *depth* (distance from the root node), *fan-out* (number of children nodes), and *density* (number of children nodes having distinct labels); factors unaddressed in existing solutions,

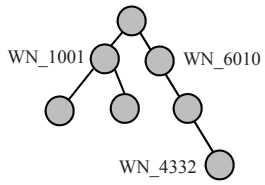
- **Module 3**: *Context definition and representation*, building the context of target nodes following the *sphere neighborhood* model introduced in [3], where contexts are represented as vectors considering a comprehensive XML structure context including all XML structural relationships within a (user-chosen) range; in contrast with partial context representations using the *bag-of-words* paradigm,




**a.** Input syntactic XML tree (representing Doc 1 in Figure 1).




**b.** Output semantic XML tree, where nodes 6 and 7 were not targeted for disambiguation.




**c.** Input semantic network (WordNet), with symbolic concepts & identifiers.

*Concept-Doc* Index

| Concept : NodeID(s) |
|---|
| C1 : 0 |
| C2 : 1 |
| … |
| $C_{14}$ : 15 |

*Concept-WN* Index

| Concept : WN_ID(s) |
|---|
| C1 : WN_1001 |
| C2 : WN_1043 |
| … |
| $C_{14}$ : WN_6010 |

**d.** Output inverted concept indexes, generated after disambiguation along with the output semantic tree.

Fig. 3. Sample input (syntactic) XML tree and output (semantic) XML trees.

- **Module 4:** *XML semantic disambiguation*, running *sphere neighborhood* vectors through a hybrid disambiguation process, combining two approaches: *concept-based* and *context-based* disambiguation methods developed in [3], allowing the user to tune disambiguation parameters following her needs; in contrast with static methods.

The *XSDF* prototype system[1] was implemented as an open software using Java. It receives as input: i) an XML document tree, ii) a reference semantic network or knowledge base (the current version uses WordNet 3.0), and iii) user parameters (to tune the disambiguation process following her needs), to produce as output a semantic XML tree (cf. Fig. 3).

The prototype also includes implementations of existing XML disambiguation methods which were used to conduct comparative experiments. A prototype snapshot is shown in Fig. 4.

## III. EXPERIMENTS

A battery of experiments was conducted to evaluate three criteria: i) *semantic ambiguity*, ii) *disambiguation quality*, and iii) *processing time*. We first briefly describe the test data, and then discuss the results.

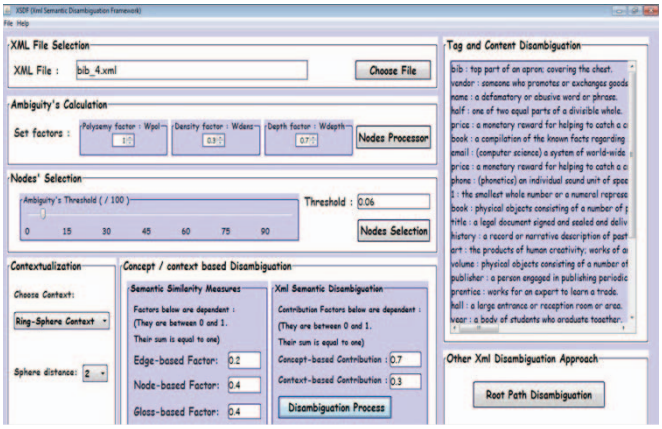



Fig. 4. Snapshot of XSDF's main interface.

TABLE I. Characteristics of test documents.

| Groups | | Datasets | N# of docs | Avg.. N# of nodes per doc | Avg. Node label polysemy | Avg. Node Depth | Avg. Node Fan-out | Avg. Node Density |
|---|---|---|---|---|---|---|---|---|
| **Group 1** | 1 | Shakespeare collection[1] | 10 | 192.054 | 7.052 | 3.687 | 0.604 | 0.38 |
| **Group 2** | 2 | Amazon product files[2] | 10 | 113.333 | 8.407 | 4.309 | 0.539 | 0.38 |
| **Group 3** | 3 | SIGMOD Record[3] | 6 | 39.375 | 4.615 | 2.743 | 0.692 | 0.692 |
| | 4 | IMDB database[4] | 6 | 15.475 | 4 | 2.666 | 1.066 | 1 |
| | 5 | Niagara collection[5] | 8 | 26.5 | 4.384 | 2.961 | 0.884 | 0.884 |
| **Group 4** | 6 | W3Schools[6] | 4 | 16.5 | 3.937 | 2.312 | 0.812 | 0.812 |
| | 7 | W3Schools | 4 | 16 | 2.375 | 2.437 | 0.562 | 0.562 |
| | 8 | W3Schools | 4 | 11.675 | 3.454 | 2 | 1.181 | 1.181 |
| | 9 | Niagara collection | 4 | 19 | 3.947 | 2.368 | 1.157 | 1.157 |
| | 10 | Niagara collection | 4 | 15.5 | 4.533 | 2.266 | 1.4 | 1.4 |

[1] Available online at http://sigappfr.acm.org/Projects/XSDF/

### A. Experimental Test Data

We used a collection of 80 test documents gathered from several data sources having different properties (cf. TABLE I), which we categorize following two features: i) *node ambiguity*, and ii) *node structure* (cf. TABLE II). The former feature highlights the average amount of ambiguity of XML nodes in the XML tree, estimated using our *ambiguity degree* measure, *Amb_Deg* ∈ [0, 1]. The latter feature describes the average amount of structural richness of XML nodes, in terms of node *depth*, *fan-out*, and *density* in the XML tree, averaged over all nodes in the XML tree, formally:

$$\text{Amb\_Deg}(x, T, SN) = \frac{w_{\text{Polysemy}} \times \text{Amb}_{Polysemy}(x.\ell, SN)}{w_{\text{Depth}} \times (1 - \text{Amb}_{Depth}(x, T)) + w_{\text{Density}} \times (1 - \text{Amb}_{Density}(x, T)) + 1} \in [0,1] \quad \textbf{(1)}$$

$$\text{Struct\_Deg}(x, T) = \frac{w_{\text{Depth}} \times x.d}{\text{Max}(depth(T))} + \frac{w_{\text{Fan-out}} \times x.f}{\text{Max}(fan\text{-}out(T))} + \frac{w_{\text{Density}} \times \overline{x.f}}{\text{Max}(\overline{fan\text{-}out(T)})} \in [0,1] \quad \textbf{(2)}$$

where $x$ is an XML node, $T$ an XML document tree, $SN$ a reference semantic network, $x.d$ the node's depth, $x.f$ the node's fan-out, $\overline{x.f}$ the node's density, $w_{Depth} + w_{Fan\text{-}out} + w_{Density} = 1$ and $(w_{Depth}, w_{Fan\text{-}out}, w_{Density}) \geq 0$. In other words, high node depth, fan-out, and/or density here indicate a highly structured XML tree, whereas low node depth, fan-out, and/or density indicate a poorly structured (relatively flat) tree. In our experiments, we set equal weights $w_{\text{Depth}} = w_{\text{Fan-out}} = w_{\text{Density}} = 1/3$ when measuring *Struct_Deg* (cf. Formula 2).

### B. XML Ambiguity Degree Correlation

We compared XML node ambiguity ratings produced by human users with those produced by our system (i.e., via our *ambiguity degree* measure, *Amb_Deg*, ), using *Pearson's correlation coefficient*, $pcc = \delta xy/(\delta x+\delta y)$ where: $x$ and $y$ designate user and system generated ambiguity degree ratings respectively, $\delta_x$ and $\delta_y$ denote the standard deviations of $x$ and $y$ respectively, and $\delta_{xy}$ denotes the covariance between the $x$ and $y$ variables. The values of $pcc \in [-1, 1]$ such that: -1 designates that one of the variables is a decreasing function of the other variable (i.e., values deemed ambiguous by human users are deemed unambiguous by the system, and visa-versa), 1 designates that one of the variables is an increasing function of the other variable (i.e., values are deemed ambiguous/unambiguous by human users and the system alike), and 0 means that the variables are not correlated. Five test subjects were involved in the experiment. Manual ambiguity ratings (integers ∈ [0, 4], i.e., ∈ [*min*, *max*] ambiguity) where acquired for 12-to-13 randomly pre-selected nodes per document, i.e., a total of 1000 nodes (during an average 10 hours rating time per tester) and then correlated with system ratings, computed

with variations of *Amb_Deg*'s parameters to stress the impact of its factors ($Amb_{Polysemy}$, $Amb_{Depth}$, and $Amb_{Density}$): i) Test #1 considers all three factors equally ($w_{Polysemy} = w_{Depth} = w_{Density} = 1$), ii) Test #2 focuses on the polysemy factor ($w_{Polysemy} = 1$ while $w_{Depth} = w_{Density} = 0$), iii) Test #3 focuses on the depth factor ($w_{Depth} = 1$ while $w_{Polysemy} = 0.2$ and $w_{Density} = 0$), iv) Test #4 focuses on the density factor ($w_{Density} = 1$, $w_{Polysemy} = 0.2$ and $w_{Depth} = 0$).

TABLE II. XML test documents organized based on average node *ambiguity* and *structure*.

| | **Structure +** | **Structure –** |
|---|---|---|
| **Ambiguity +** | **Group 1**<br>*Amb_Deg* = 0.1127 &<br>*Struct_Deg* = 0.6803 | **Group 2**<br>*Amb_Deg* = 0.1378 &<br>*Struct_Deg* = 0.6621 |
| **Ambiguity –** | **Group 3**<br>*Amb_Deg* = 0.0625 &<br>*Struct_Deg* = 0.612 | **Group 4**<br>*Amb_Deg* = 0.0447 &<br>*Struct_Deg* = 0.5515 |

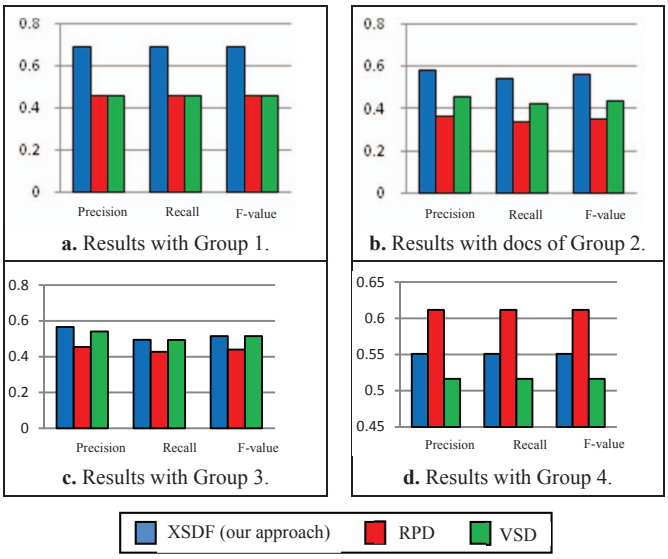
TABLE III. Correlation between human ratings and system generated ambiguity degrees (cf. detailed graphs in [17]).

| | | **Test #1** *All factors* | **Test #2** *Polysemy* | **Test #3** *Depth* | **Test #4** *Density* |
|---|---|---|---|---|---|
| **Group 1** | Dataset 1 | 0.394 | 0.411 | 0.335 | **0.439** |
| **Group 2** | Dataset 2 | **0.017** | 0.181 | 0.243 | 0.139 |
| **Group 3** | Dataset 3 | -0.087 | -0.139 | -0.071 | -0.138 |
| | Dataset 4 | 0.408 | 0.438 | 0.390 | 0.398 |
| | Dataset 5 | -0.184 | -0.185 | -0.131 | -0.235 |
| **Group 4** | Dataset 6 | -0.284 | -0.291 | -0.243 | -0.316 |
| | Dataset 7 | -0.177 | -0.190 | -0.254 | -0.143 |
| | Dataset 8 | -0.119 | -0.025 | 0.033 | -0.156 |
| | Dataset 9 | -0.452 | -0.301 | -0.251 | **-0.456** |
| | Dataset 10 | -0.258 | 0.180 | 0.412 | 0.276 |

Results compiled in TABLE III highlight several observations. First, XML ambiguity seems to be perceived and evaluated similarly by human users and our system – obtaining maximum positive correlation between human and *Amb_Deg* scores – when highly ambiguous and highly structured XML nodes are involved (e.g., Group 1). Second, ambiguity seems to be evaluated differently by users and our system when less ambiguous and/or poorly structured XML nodes are involved, attaining: negative or close to null correlation when low ambiguity and/or poorly structured XML nodes are evaluated (e.g., Groups 2, 3, and 4). This is due to the intuitive understanding of semantic meaning by humans, in comparison with the intricate processing done by our automated system. For instance, in the case of documents of Data-set 9 of Group 4, the meaning of child node label "*state*" under node label "*address*" was obvious for our human testers (providing an ambiguity score of 0/4). Yet, "*state*" has 8 different meanings following WordNet and thus its meaning is not so obvious for a machine. Concerning *Amb_Deg*'s parameter weight variations (for $w_{Polysemy}$, $w_{Depth}$, and $w_{Density}$) with tests 2, 3, and 4, all three parameters seem to have comparable impacts on ambiguity evaluation. Note that evaluating XML node ambiguity is dismissed in existing approaches, since they do not address the issue of selecting target (ambiguous) nodes for disambiguation (they simply disambiguate all nodes in an XML tree, which can be complex and needless).

### C. XML Disambiguation Quality

In addition, we evaluated the disambiguation quality of our approach in comparison with two of its recent alternatives: *RPD* (Root Path Disambiguation) [14], and *VSD* (Versatile Structure Disambiguation) [10]. A qualitative comparison is shown inTABLE IV. We ran a battery of tests considering the different features and configurations or our approach.



**a.** Results with Group 1.
**b.** Results with docs of Group 2.
**c.** Results with Group 3.
**d.** Results with Group 4.

Fig. 5. Average *PR*, *R* and *F-value* scores comparing our approach with RPD [14] and VSD [10].

Results in Fig. 5 show that our method yields *precision, recall,* and *f-value* levels higher than those achieved by its predecessors, with almost all test groups except with Group 4 (cf. Fig. 5.d) where *RPD* produces better results. In fact, XML nodes in Group 4 are less ambiguous and poorly structured in comparison with remaining test groups. Hence, choosing a simple context made of root path nodes has proven to be less noisy in this case (including less context nodes) in comparison with the more comprehensive context models used with our approach and with *VSD*. One can also realize that our method produces highest *precision*, *recall*, and *f-value* levels with Group 1 (*high ambiguity* and *rich structure* XML nodes), with an average 35% improvement over *RDP* and *VSD* (cf. Fig. 5.a), in comparison with average 25%, 5%, and almost 0% improvements with Groups 2, 3, and 4 respectively (cf. Fig. 5.b, c, d). This concurs with our results of the previous section: our method is more effective when dealing with highly ambiguous nodes within a rich XML structure, in comparison with less ambiguous/poorly structured XML.

TABLE IV. Comparing *XSDF* with existing approaches

| Approaches | Considers linguistic pre-processing | Considers tag tokenization (compound terms) | Addresses XML node ambiguity | Integrates an inclusive XML structure context | Flexible w.r.t. context size | Adopts *relational information* approach | Combines the results of various semantic similarity measures | Straightforward mathematical functions | Disambiguates XML structure and content |
|---|---|---|---|---|---|---|---|---|---|
| RPD [50] | √ | x | x | X | x | x | x | x | x |
| VSD [29] | √ | √ | x | √ | √ | √ | x | x | x |
| **XSDF** (our approach) | **√** | **√** | **√** | **√** | **√** | **√** | **√** | **√** | **√** |

### *D. Performance Evaluation*

We have also conducted a detailed complexity analysis and various time experiments to study and evaluate the performance of our approach. Comparative results in Fig. 6 show that our method is of average polynomial complexity, and that its time performance is closely comparable to those of its alternatives (cf. details in [17]).

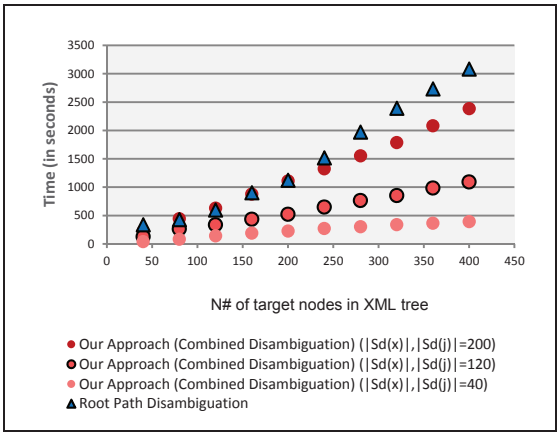



Fig. 6. Comparing time results with existing *RPD* [14] approach.

In the oral demonstration, we aim to showcase our system's effectiveness and efficiency in disambiguating XML data, while emphasizing its logical design, implementation, and functionality.

## REFERENCES


[1] Aleman-Meza B. *et al.*, *Scalable Semantic Analytics on Social Networks for Addressing the Problem of Conflict of Interest Detection.* ACM TWeb, 2008. 2(1):7.

[2] Berendt B. *et al.*, *Bridging the Gap - Data Mining and Social Network Analysis for Integrating Semantic Web and Web 2.0.* Journal of Web Semantics, 2010. 8(2-3): 95-96.

[3] Charbel N. *et al.*, *Resolving XML Semantic Ambiguity.* EDBT' Conf., 2015. Brussels, pp 277-288.

[4] Cimiano P. *et al.*, *Towards the Self-Annotating Web.* WWW Conf., 2004. pp. 462-471.

[5] Do H. and Rahm E., *Matching Large Schemas: Approaches and Evaluation.* Information Systems, 2007. 32(6): 857-885.

[6] Fellbaum C., *Wordnet: An Electronic Lexical Database.* MIT Press, 1998. 422 p.

[7] Ide N. and Veronis J., *Introduction to the Special Issue on Word Sense Disambiguation: The State of the Art.* Computational Linguistics, 1998. 24(1):1-40.

[8] Krovetz R. and Croft W. B., *Lexical Ambiguity and Information Retrieval.* ACM Transactions on Information Systems, 1992. 10(2):115-141.

[9] Maguitman A. *et al.*., *Algorithmic Detection of Semantic Similarity.* WWW Conf., 2005. pp. 107-116.

[10] Mandreoli F. and Martoglia R., *Knowledge-based sense disambiguation (almost) for all structures.* Information Systems, 2011. 36(2): 406-430.

[11] Navigli R., *Word Sense Disambiguation: a Survey.* ACM Compt. Surveys, 2009. 41(2):1–69.

[12] Navigli R. and Velardi P., *An Analysis of Ontology-based Query Expansion Strategies.* In proc. of the Inter. Joint Conferences on Artificial Intelligence (IJCAI'03), 2003. pp. 42-49.

[13] Tagarelli A. and Greco S., *Semantic Clustering of XML Documents.* ACM TOIS, 2010. 28(1):3.

[14] Tagarelli A. *et al.*, *Word Sense Disambiguation for XML Structure Feature Generation.* In Proceedings of the European Semantic Web Conference, 2009. LNCS 5554, pp. 143–157.

[15] Taha K. and Elmasri R., *CXLEngine: A Comprehensive XML Loosely Structured Search Engine.* Proc. of the EDBT DataX workshop, 2008. pp. 37-42, Nantes, France.

[16] Taha K. and Elmasri R., *XCDSearch: An XML Context-Driven Search Engine.* IEEE TKDE, 2010. 22(12):1781-1796.

[17] Tekli J. *et al.*, *A Dynamic Framework for XML Semantic Disambiguation.* Submitted to Elsevier Journal of Web Semantics (JWS), 2015.

[18] Tekli J. *et al.*, *A Novel XML Structure Comparison Framework based on Sub-tree Commonalities and Label Semantics.* Elsevier J. of Web Semantics (JWS):, 2012. 11: 14-40.

[19] Tekli J. *et al.*, *Minimizing User Effort in XML Grammar Matching.* Elsevier Info. Sci., 2012, 210:1-40.

[20] Theobald M. *et al.*, *Exploiting Structure, Annotation, and Ontological Knowledge for Automatic Classification of XML Data.* ACM SIGMOD WebDB, 2003. pp. 1-6

[21] Yaworsky D., *Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora.* In proc. of COLING, 1992. Vol 2, pp. 454-460. Nantes.