PDTB XML: the XMLization of the Penn Discourse TreeBank 2.0

Xuchen Yao, Irina Borisova, Mehwish Alam

Faculty of Arts
University of Groningen
{yaoxuchen,borisova.ira,malam.ravian}@gmail.com

Abstract

The current study presents a conversion and unification of the Penn Discourse TreeBank 2.0 under the XML format. The converted corpus allows for a simultaneous search for syntactically specified discourse information on the basis of the XQuery standard.

1. Introduction

In Natural Language Processing (NLP), linguistic data is frequently reused, while the reuse of technologies remains relatively rare. The reasons for this are twofold. First, in technological issues (such as the use of different implementation languages), preferences for environments and paradigms are diverse. Second, the developed applications are often theory-dependent, that is, explicitly representing various underlying theories (Leidner, 2003). In order to be reapplied, a technological platform or an encoding format has to satisfy very general requirements and at the same time permit significant modifications.

Recently, XML has become widely used in various NLP applications. Its contribution to tasks in linguistic field is largely based on the benefits of a standard and flexible language. XML offers a clear and consistent data structure: for novice users, who don't have any experience of working with analyzed data, learning it is quite simple. Flexible data organization allows a user to add and define the document structure depending on his needs. Finally, its consistent encoding format facilitates the combinations of different corpora, and, furthermore, a unified query language.

There are numerous examples of the implementation of XML in corpus annotation and alignment tasks. Among them are the Alpino Dependency Treebank (Bouma and Kloosterman, 2002), the Europarl parallel corpus (Koehn, 2005), the Wikipedia XML corpora (Denoyer and Gallinari, 2006; Schenkel et al., 2007), and others. Other studies aim to convert data into XML corpora (Grover et al., 2002; Huhmarniemi et al., 2007), and to make XML representations for uniting the corpus data of multiple sources (Volk et al., 2006). The XML Corpus Encoding Standard (XCES), released by the Expert Advisory Group on Language Engineering Standards, states the minimal encoding and annotation requirements in ready-to-use XCES Schemas and DTDs[1].

PDTB XML is such a project that converts the Penn Discourse Treebank 2.0 (PDTB, (Prasad et al., 2008)) into XML format. The PDTB is a large-scale linguistic corpus manually annotated with discourse relations, arguments, attributions and senses. These relations are lexically related to their syntactic structures (by including the Penn Treebank (PTB, (Marcus et al., 1993))). The PDTB functions as a valuable resource for discourse analysis and other natural language applications, such as question answering and natural language generation.

As thoroughly annotated linguistic corpora, the PTB and the PDTB offer a number of important benefits for research in linguistics and information technologies. Both corpora follow a theory-neutral approach: both annotations are lexically-based, that is, the discourse connectives in the PDTB are tagged as explicit representations of the PDTB relations, and the hierarchical sense annotation is relevant for the PTB and PDTB corpora. However, two main difficulties in using the corpora may be observed:

- Users cannot query the PDTB and the PTB simultaneously, so they must perform one query on discourse relations first and then on syntactic structure with the results from the first step, or vice versa. This limitation is due to the restricted functionalities of the PDTB API.
- Since the PDTB uses a very specific file format, users cannot easily add more annotations to the corpus. Otherwise, the PDTB API will not work properly due to a broken file format. [2] In this sense, the PDTB is not easily extendable by the users.

The PDTB XML project aims to answer these user demands by converting the PDTB into XML format and integrating it with the PTB, also converted in XML. Given a reasonable XML format, one can use the standard XML query language, XQuery, to extract information from PDTB XML. All the conversion tools are

[1] `http://www.xces.org/`

[2] A new annotation tool (`http://www.seas.upenn.edu/~pdtb/PDTBAPI/Annotator.html`) was released to help annotate corpora with a graphic user interface. But this tool creates two additional files for each raw text file: one stores the annotation records and the other stores comments. Moreover, at the time of writing, the original PDTB files could not be changed with the annotation tool.

released in the public domain[3].
In the development of PDTB XML, other XML-based formats and tools are also taken into consideration. (Dipper, 2005) suggests a generic XML-based stand-off architecture for linguistic annotation, but it does not take into account any immediate connection between the initial data and its annotation. This approach does not explain the search procedure within the annotations and especially the issue of accessing the shared information. The stand-off architecture might have to undergo many changes in order to use it for application-specific needs. The Graph Annotation Format (GrAF) introduced by (Ide and Suderman, 2007) aims to represent merged linguistic annotations as a single connected mapped graph. This architecture gives an insight into how the structures in treebanks are defined in terms of graphical representation, which is an extension to the Linguistic Annotation Framework (Ide and Romary, 2006). The final graph diagrams are generated by GraphViz(Gansner and North, 1999). However, graphics generated from GraphViz are static, so they lack user interaction and can only be used for display purposes. Also, a mass production of graphics from different corpora is usually too costly.
The NITE XML Toolkit (NXT, (Carletta et al., 2003)) is mainly used in human interaction research. For instance, it can simultaneously show linguistic annotations from multiple sources and multiple models in a meeting scenario. Obviously, this task does not serve the needs of PDTB XML. An XML format focusing on the structure and content rather than timings of a corpus should do. As will be explained later, the Tiger XML format is more suitable in this case.
The paper is organized as follows. Section 2 gives a short introduction of PDTB and Section 3 details the procedure and designing protocols of conversion. Section 4 uses a simple example to illustrate how a query is done in PDTB XML and section 5 presents a final conclusion and proposes future work.

2. The Penn Discourse TreeBank

The Penn Treebank (PTB) and the Penn Discourse TreeBank (PDTB) are valuable resources for syntactic and discourse analyses. The PTB consists of more than 4.5 million words from the texts of the Wall Street Journal corpus, the Brown corpus, and other sources. They are annotated with part-of-speech tags and more than half of them with syntactic information.
The PDTB is a large-scale linguistic corpus manually annotated with discourse relations, arguments, attributions and senses. It presents encoded information about textual coherence relations that are classified by the discourse connectives and lexically related to their syntactic structures. The PDTB defines five relations: *Explicit, Implicit, AltLex, EntRel,* and *NoRel.* A connective can have only two arguments, *Arg1* and *Arg2*, derived by their positions in relation to the connective. In addition, arguments may have two supplements, defined only in cases where they are relevant for the PDTB relations and tagged as *Sup1* and *Sup2*. An example of explicit relations is given below.
Explicit relation is formed by the presence of an explicit connective, one that belongs to a defined syntactic class. For instance "then" is an explicit connective in the sentence below (the connective is underlined, *Arg1* is in underlined italics, **Arg2** is in bold and *Sup1* is in in italics.)

(1) *A buffet breakfast was held in the museum*, *where food and drinks are banned to everyday visitors.* Then, **in the guests' honor, the speedway hauled out four drivers, crews and even the official Indianapolis 500 announcer for a 10-lap exhibition race**.

3. Conversion into XML

The conversion process includes three procedures: the XMLization of syntactic structures (the PTB), the XMLization of discourse relations (the PDTB), and the combination of these two parts into one single XML file.

3.1. PTB XML

TIGER-XML (Brants et al., 2002) is used to represent the PTB syntactic trees. TIGER-XML is a representation format for syntax tree structures, with nodes, edge labels, multiple features of words and crossing edges. Two main benefits of the TIGER framework are its independence of any linguistic theories that often underlie the corpus encoding format, and its representation format. Also, TIGER-XML criteria are very general and thus support a wide range of existing formats. The PDTB XML follows these principles and uses TIGER-XML to encode the syntactic parts of the corpus. A stand-alone converter is extracted from the original TIGERRegistry[4] software to convert the parse trees of the PTB (`.mrg` files) into TIGER-XML format. Thus, this part of XML is compatible with TIGER-XML format. The parse trees can also be shown by the TIGERSearch software properly.

3.2. PDTB XML

For the discourse relations, a new XML encoding has been developed. The general design of the relations in XML format adheres to the following principles:

1. All five types of relations should have similar structures to assure that general queries can be performed on all relations regardless of their type.
2. Entities inside a discourse relation should have a reasonable XPath structure.
3. References to syntax should be such that XQuery can be used to find the corresponding syntactic structures.

[3] `http://code.google.com/p/pdtb-xml/`

[4] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/`

The first principle guarantees a unified format for the convenience of XQuery users, the second principle integrates discourse relations and XML structures into one unified hierarchy and the third principle builds the cross references between the syntactic trees and the discourse relations.
The PDTB uses Gorn addresses for referring the relations to the syntax fragments. For instance, a Gorn address such as `1,2,3,4` denotes the 4th child of the 3rd child of the 2nd child of the 1st tree in a PTB bracketing format. This format is not very convenient for the XML-based search, so the Gorn addresses were substituted by the explicit IDs for the syntax trees and the references to these IDs for the discourse parts.
Figure 1 gives an example of the Explicit relation format. Elements in italics (such as *Sup1*, *Attribution*) are optional.
The features of relations, arguments and connectives are coded as attributes[5] in the PDTB XML format.

3.3. Building Cross References

The syntactic and discourse elements were assigned with unique IDs in order to build the cross references between them:

1. `<s>` element for sentence (such as `"s1"`, the first sentence in the file).
2. `<t>` element for terminals (such as `"s1_2"`, the second word of the first sentence).
3. `<nt>` element for nonterminals (such as `"s1_500"`, internal coding).
4. `<tree>` element for trees (such as `"t1"`, the first tree).
5. `<b>` element for branches in trees (such as `"t1_2_3"`, the 3rd child of the 2nd child of the 1st tree).
6. `<Relation>` element for relations (such as `"r1"`, the first relation in the file).

If an element needs to refer to some other elements, an `idref` attribute is added to that element for the reference. This enables queries to trace back and forth between different elements. Parse trees in the PDTB XML are represented by the `<tree>` element, which is attached to every sentence by traversing and expanding all the nonterminals and terminals. Relations, on the other hand, are equipped with `<TreeRef>` elements referring to the matching parse trees.

3.4. Combined PDTB XML

An XSLT style sheet is used to combine the PTB and PDTB XML files into one file. The final XML file contains a single `<corpus>` element, which has three child elements: `<head>` (general info), `<body>` (the PTB part) and `<Relations>` (the PDTB part), as shown in Figure 2.

3.4.1. Validation

The final XML files are defined by a set of XML schemata[6] and, consequently, can be validated by these schemata. This is a compulsory process after conversion: it is essential in order to make sure that every converted XML file conforms to the same standard.

4. Querying the PDTB XML

One of the most important results of the conversion and combination of the PDTB and the PTB is the simultaneous search for the syntactic information that underlies the discourse relations, and vice versa. Querying the part-of-speech tags of the discourse connectives of different semantic classes is an example of such task. For instance, the connectives within *Contrast*, a subclass of the *Comparison* semantic class, might be either coordinating conjunctions, subordinating conjunctions, adverbs, or a combination of two. In PDTB API this cannot be done in one step: one has to first query the discourse part, save it, then query the syntactic part. In the PDTB XML this task can be solved within one combined query (based on XQuery language).
XQuery is a standard query language for XML, specifically designed for retrieving and interpreting different XML data sources. Its syntax is based on XPath, a navigating query language for the search of the data within the XML document. XQuery is applied for testing the PDTB XML functionality; Figure 3 shows a query written to find the POS tags of the connectives belonging to the *Comparison.Contrast* semantic class (also cf. Figure 1 and 2).
The `collection()` function retrieves all the XML documents inside a directory and the `for` loop transforms all the `<Relation>` elements into sequences, which are processed one by one inside the `for` loop. The slash operator / walks through the top of a document down to the `<Relation>` element. The @ symbol refers to attributes of an element and restricts the element by residing inside a pair of brackets ([ ]). Thus `$rel` at line 1 of Figure 3a is assigned to the *<Relation>* element (at line 1 of Figure 3b) whose *<Connective>* element's *SemanticClass1* attribute is *Comparison.Contrast*. `$ref/@idref` at line 4 of Figure 3a returns the referenced ID *t4_1_1* at line 5 of Figure 3b. An XQuery function `id("t4_1_1")` returns the element whose ID is *t4_1_1*, i.e. the *<b>* element at line 10 of Figure 3b, whose POS tag is *IN*.
Similarly, if we have known a syntactic tag *<b>* which has an ID of *t4_1_1*, the XQuery function `idref("t4_1_1")` returns all elements referring *t4_1_1*. By encoding cross references between syntactic trees and discourse relations, PDTB XML is capable of querying both parts simultaneously.

[5] Tab29-31Col1 in Figure 1 reads: Table 29 to Table 31, Column 1, in the PDTB annotation manual: `http://www.seas.upenn.edu/~pdtb/PDTBAPI/pdtb-annotation-manual.pdf`

[6] `http://code.google.com/p/pdtb-xml/source/browse/#svn/trunk/schema`

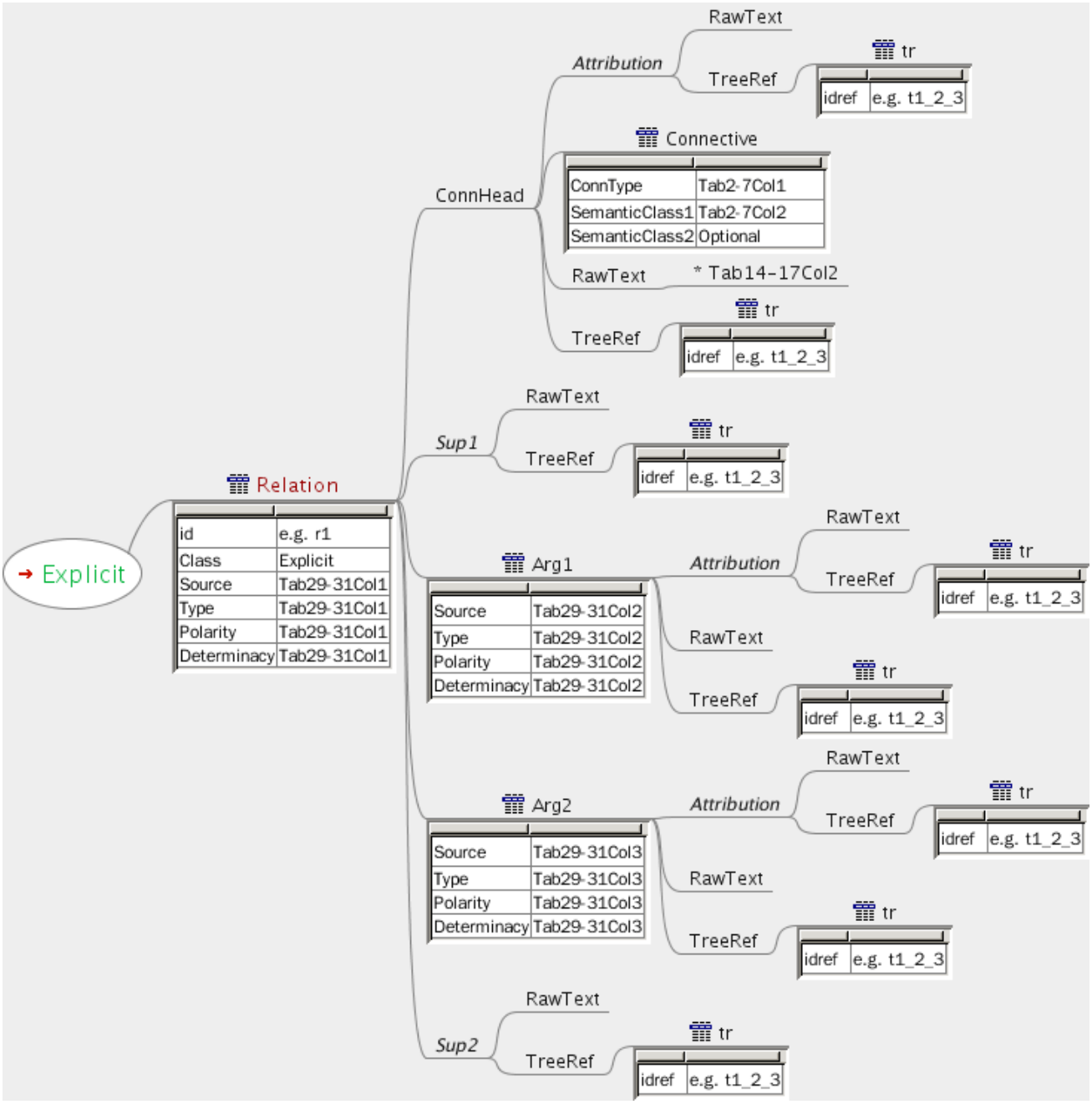

Figure 1: Explicit relation hierarchy in XML format

Note that the PDTB XML guarantees the well-formedness and validity of converted XML by testing the files on a set of predefined XML schemata. It is just a different form of data representation and it does not eliminate any errors existing in the original data, nor does it introduce new errors. For instance, the research carried by (Dines et al., 2005) regards the issue of the merging syntactic and discourse annotations in the PTB and the PDTB with respect to the mismatch of the annotations in the arguments of subordinate conjunctions (i.e. the cases when `Arg1/2` in the PTB contains more information than its annotation in the PDTB). The authors suggest a tree-subtracting algorithm, applied for extracting the arguments in the PTB, with the possible classification of the material as exact, extra, and omitted. With the same experiment conducted on PDTB XML, it should give the same result of classification and mismatch. Importantly, this study is based on the evaluation of the discourse annotation, whereas in our work we have not raised the question of the correctness of data annotation.

5. Conclusions and Future Work

A joint XML corpus of the PTB and the PDTB is created in order to simplify the search procedure and to broaden the search possibilities. The key stages here include developing proper cross-references between different data types and their representation in the modified TIGER-XML format, and then writing the required declarative languages (XML Schema). Note,

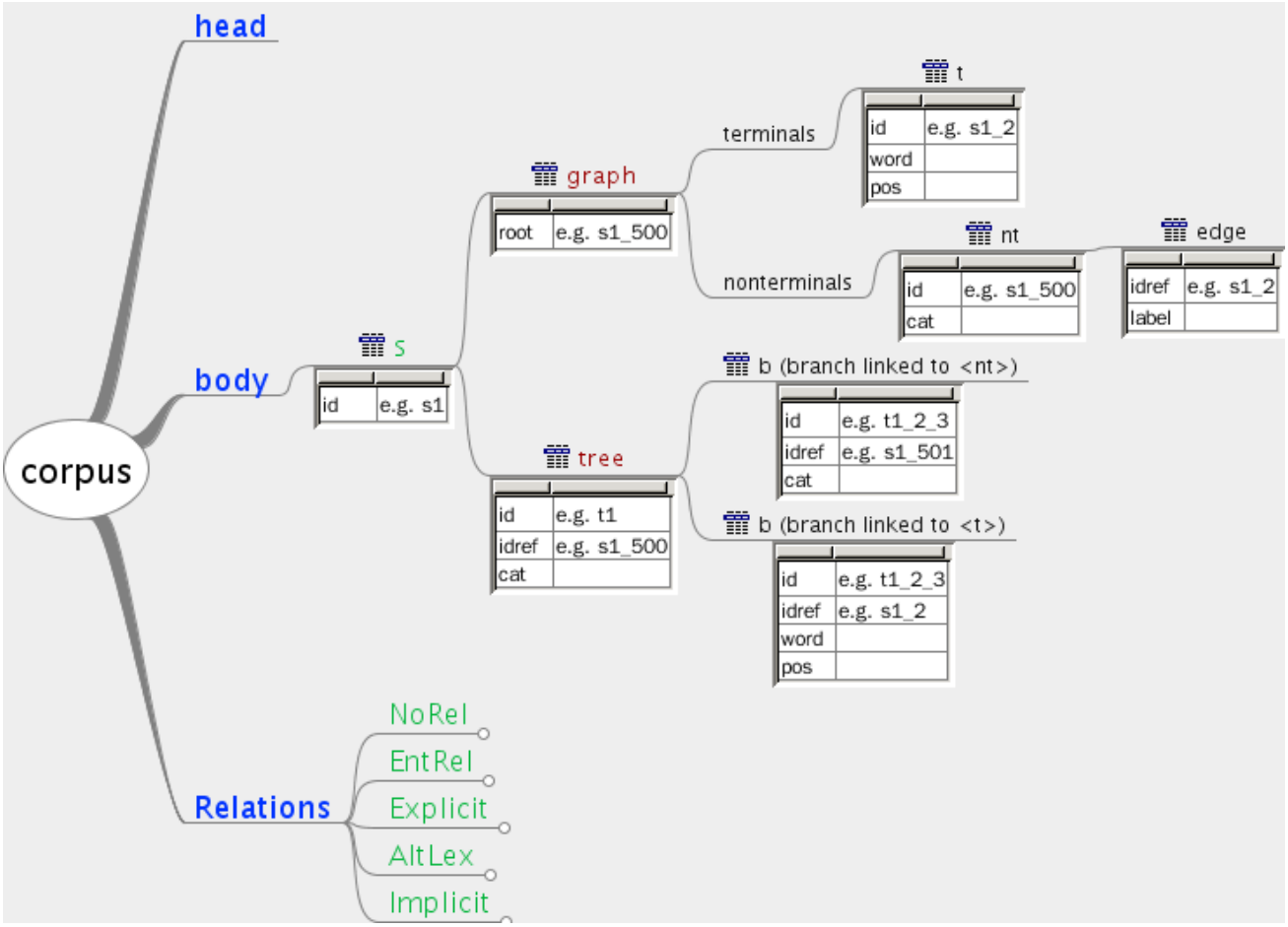

Figure 2: Top-elements in the PDTB XML file

```
for $rel in collection("xml-dir")/corpus/Relations/Explicit/Relation
[ConnHead/Connective[@SemanticClass1="Comparison.Contrast"]]
  let $connective := normalize-space(string($rel/ConnHead/RawText))
  let $ref := $rel/ConnHead/TreeRef/tr
  let $syntax_node := $ref/ancestor::*//id($ref/@idref)
  let $pos := $syntax_node/descendant-or-self::*/@pos
  return <rel connective="{$connective}" pos="{$pos}"/>
```

(a) the XQuery code

```
<Relation id="r3" Class="Explicit" Source="Wr" Type="Comm" Polarity="Null" Determinacy="Null">
  <ConnHead>
    <Connective ConnType="although" SemanticClass1="Comparison.Contrast"/>
    <RawText>Although</RawText>
    <TreeRef> <tr idref="t4_1_1"/>
</TreeRef>
  </ConnHead>
  ......
</Relation>
  ......
<b id="t4_1_1" idref="s4_1" word="Although" pos="IN"/>
```

(b) a sample XML excerpt

Figure 3: An XQuery example to find all connectives belonging to the *Comparison.Contrast* semantic class and their POS tags.

however, the PDTB XML is designed as a supplement to the PDTB, rather than a replacement. It still lacks an interactive graphic user interface and has not been tested on large-scale queries. Thus, after conversion, the PDTB users can have both the benefits of a GUI, query functionalities from the PDTB API, and extensibility and the standard format of XQuery language from the PDTB XML.

We believe that the presented technique of XMLization of the different data types can be further implemented as a template for developing new corpora. In order to move in this direction, large-scale XQuery searches should be performed on the data. This is essential for checking or revising the effectiveness of the representation power of the XML structures. Also, to ease the work of the PDTB researchers, a set of XQuery APIs is under development that will assist users in managing common search tasks.

6. Acknowledgements

This project was accomplished as a part of our studies within the Erasmus Mundus European Masters Program in Language and Communication Technologies. The authors are very grateful to Gosse Bouma, Gisela Redeker, Jennifer Spenader, and the students in the PDTB research group at the University of Groningen, The Netherlands, for their supervision, inspiring interest and valuable suggestions. We would like to thank the anonymous reviewers for the careful reading and very useful comments.

7. References

Gosse Bouma and Geert Kloosterman. 2002. Querying Dependency Treebanks in XML. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.

Jean Carletta, Stefan Evert, Ulrich Heid, Jonathan Kilgour, Judy Robertson, and Holger Voormann. 2003. The NITE XML Toolkit: flexible annotation for multimodal language data. *Behavior Research Methods Instruments and Computers*, 35(3):353–363.

Ludovic Denoyer and Patric Gallinari. 2006. The Wikipedia XML Corpus. *ACM SIGIR Forum*, 40(1):64–69.

Nikhil Dines, Alan Lee, Eleni Miltsakaki, Rashmi Prasad, Aravind Joshi, and Bonnie Webber. 2005. Attribution and the (non-)alignment of syntactic and discourse arguments of connectives. In *CorpusAnno '05: Proceedings of the Workshop on Frontiers in Corpus Annotations II*, pages 29–36, Morristown, NJ, USA. Association for Computational Linguistics.

Stefanie Dipper. 2005. XML-based stand-off representation and exploitation of multi-level linguistic annotation. In *Proceedings of Berliner XML Tage 2005 (BXML 2005)*, page 39–50, Berlin.

Emden R. Gansner and Stephen C. North. 1999. An Open Graph Visualization System and Its Applications to Software Engineering. *Software - Practice and Experience*, 30:1203–1233.

Claire Grover, Ewan Klein, Mirella Lapata, and Alex Lascarides. 2002. XML-Based NLP Tools for Analysing and Annotating Medical Language. In *Proceedings of the Second Workshop on NLP and XML, September 01, 2002*, pages 1–8.

Saara Huhmarniemi, Sjur Moshagen, and Trond Trosterud. 2007. Usage of XSL Stylesheets for the Annotation of the Sami Language Corpora. In *Proceedings of the Linguistic Annotation Workshop, Prague, June 2007*, pages 45–48.

Nancy Ide and Laurent Romary. 2006. Representing linguistic corpora and their annotations. In *Proceedings of the Fifth Language Resources and Evaluation Conference (LREC)*.

Nancy Ide and Keith Suderman. 2007. Graf: a graph-based format for linguistic annotations. In *LAW '07: Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Morristown, NJ, USA. Association for Computational Linguistics.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit X (MT'05)*, pages 79–86.

Jochen L. Leidner. 2003. Current Issues in Software Engineering for Natural Language Processing. In *Proceedings of the Workshop on Software Engineering and Architecture of Language Technology Systems (SEALTS), the Joint Conference for Human Language Technology and the Annual Meeting of the Noth American Chapter of the Association for Computational Linguistics*, pages 45–50.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May.

Ralf Schenkel, Fabian M. Suchanek, and Gjergji Kasneci. 2007. YAWN: A Semantically Annotated Wikipedia XML Corpus. *BTW*, pages 277–291.

Martin Volk, Sofia Gustafson-Capkova, Joakim Lundborg, Torsten Marek, Yvonne Samuelsson, and Frida Tidstrom. 2006. XML-based Phrase Alignment in Parallel Treebanks. In *Proceedings of EACL Workshop on Multi-dimensional Markup in Natural Language Processing*.