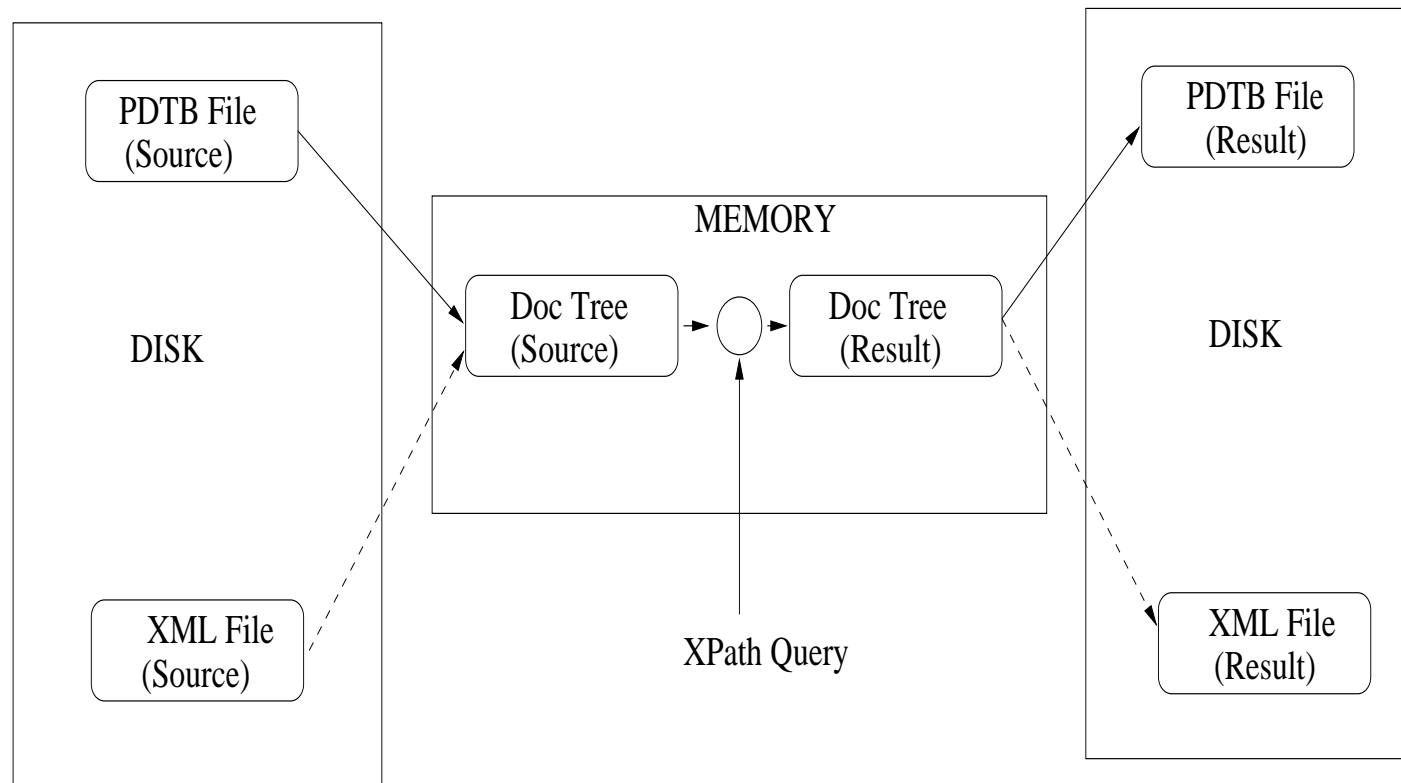


The PDTB XPath Tool

A limited mechanism to query the PDTB annotations from the command line

System Overview



The PDTB File

☞ A list of relations. Each relation is associated with:

⇒ “Attributes” which don’t have too much structure.

Examples: the head of the connective, the span of the raw text of the connective, attribution etc.

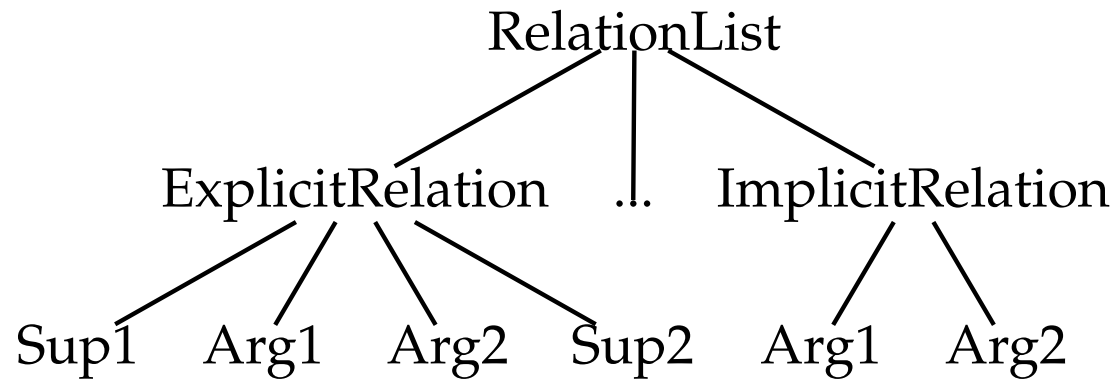
⇒ “Children” which have some structure

Arguments and supplementary information

⇒ The children in turn have attributes, but no children.

The divide between attributes and children is an artifact of the tool used in annotation, carried over to the browser display

The Document Tree



John will go to the party even if Mary doesn't

- ➡ ExplicitRelation[connHead="if" rawText="even if"]
- ➡ Arg1[rawText="John will go to the party "]
- ➡ Arg2[rawText="Mary doesn't "]
- ➡ ImplicitRelation[conn1="because" semanticClass1="Cause"
conn2="for example" semanticClass2="Add.Info"]
- ➡ Webpage (last slide) gives a complete list

XPath Queries

☞ Two syntactic devices:

⇒ Axes - ancestor, descendant, following, preceding etc.

⇒ Functions - contains, position, count, equality, comparison etc.

☞ Semantics is defined w.r.t. a “context node”

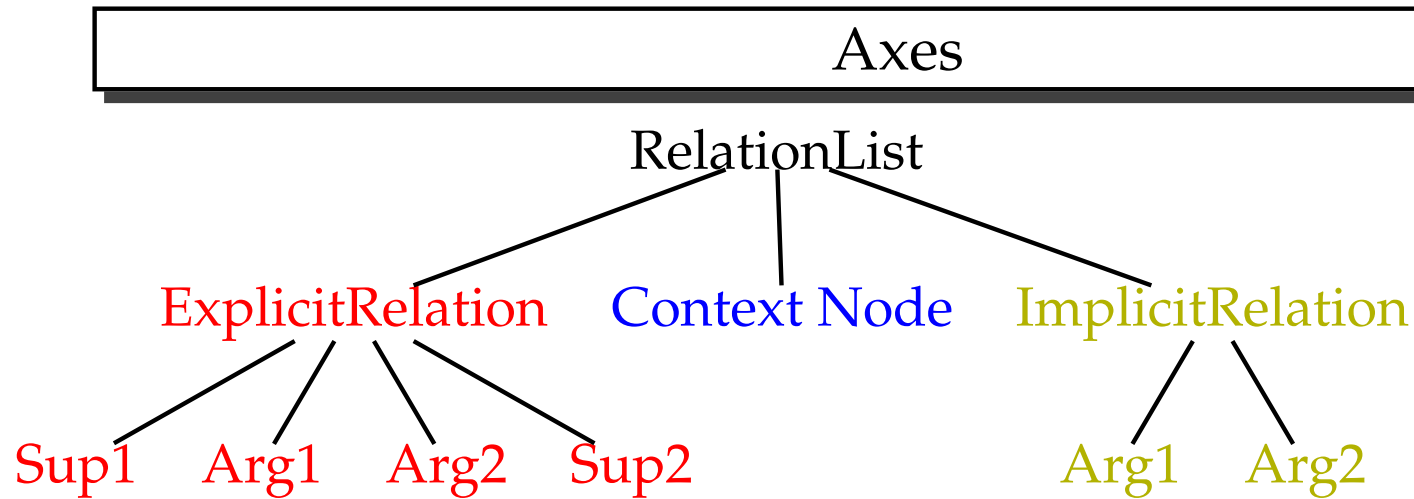
⇒ Axes select a list of nodes in relation to a context

⇒ Functions take some explicit arguments and the context implicitly.

⇒ Functions return different kinds of values, e.g., booleans, strings, nodes etc.

☞ descendant::ExplicitRelation[@rawText="if"]

Typical form of queries: Axis::NodeName[Boolean Function]



Ancestor Axis

Preceding Axis

Following Axis

Other axes: Descendant, Child, Preceding-Sibling, Following-Sibling

Example Queries

All instances of a particular connective

```
descendant::ExplicitRelation[@connHead="because"]
```

Abbreviation:

```
descendant::*[@connHead="because"]
```

Further Abbreviation:

```
//*[@connHead="because"]
```

- ➔ Running the tool
- ➔ Perl script "run_query.pl" from the webpage (last slide)
- ➔ Modify directories RawRoot PTBRoot PDTBRoot
- ➔ Usage: run_query.pl OutputDir XPathQuery
- ➔ run_query.pl QOut "//*[@connHead='because']"

More Examples

➔ Modified Connectives:

```
/*[@connHead='if' and contains(@rawText,'even')]
//ExplicitRelation[regexp(@rawText,'.*even.*if.*')]
```

➔ Sentence Intial Connectives:

```
/*[@connHead='also' and contains(@rawText,'Also')]
```

➔ Text in argument

```
/*[@connHead='if' and
  child::Arg2[regexp(@rawText,'.*\\W*not\\W*.*)']]
```

More Examples

➔ Relative Position:

```
/**[(@connHead='if' and
    following-sibling[@connHead='otherwise' and
        position()=1]) or
    (@connHead='otherwise' and
    preceding-sibling[@connHead='if' and
        position()=1])]
```

➔ Note on multi-word connHead

```
/**[@connHead='if then']
```

➔ Attribute Axis

```
//ImplicitRelation[attribute::*[string()='Cause']]
```


Notes

- ➔ System Requirements: Java 1.4.2
- ➔ Perl - if you intend to use the script
- ➔ Bottom left window of browser doesn't work on Macs
- ➔ Queries on the syntax - API, Java + XPath
- ➔ The XPath mechanism is easily extensible
- ➔ Contributions welcome

Resources

PDTB XPath Tool:

<http://www.seas.upenn.edu/~nikhild/PDTBAPI/#pdtbxpath>

XPath reference:

<http://www.w3.org/TR/xpath>

Regular Expressions:

<http://java.sun.com/j2se/1.4.2/docs/api/java/util/regex/Pattern.html>

XPath Engine:

www.jaxen.org

Bugs/Questions:

Nikhil Dinesh nikhild@seas.upenn.edu

Alan Lee aleewk@babel.ling.upenn.edu