OntoNotes: The 90% Solution

Sameer S Pradhan\textsuperscript{1}  Nianwen Xue\textsuperscript{2}

\textsuperscript{1}BBN Technologies, Cambridge, MA  
\textsuperscript{2}Brandeis University, Waltham, MA

HLT/NAACL 2009, Boulder, Colorado
The OntoNotes Project

- The OntoNotes Project started in 2006 and is a collaboration between
  - BBN Technologies
    - Ralph Weischedel
    - Lance Ramshaw
    - Sameer Pradhan
  - Brandeis University
    - Nianwen Xue
  - University of Colorado
    - Martha Palmer
  - University of Pennsylvania, and
    - Mitch Marcus
  - USC’s Information Sciences Institute
    - Eduard Hovy
    - Robert Belvin
What is OntoNotes? (I)

- Multiple layers of annotation
  - Syntax
  - Propositions
  - Word sense
  - Coreference
  - Names
  - Ontology

- Multilingual resource
  - English
  - Chinese
  - Arabic

- Parallel Data
What is OntoNotes? (II)

- Skeletal representation of literal meaning
- Find “sweet spot”
  - In depth of representation
  - Inter-Annotator Agreement (~90%)
  - Productivity
- Integrated Representation
- API for ease of use
- Distribute data widely through LDC
The Founder of Pakistan’s nuclear department, Abdul Qadeer Khan, has admitted he transferred nuclear technology to Iran, Libya and North Korea.
## Amount of Data

### OntoNotes 3.0

<table>
<thead>
<tr>
<th>Language</th>
<th>NW</th>
<th>BN</th>
<th>BC</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>500</td>
<td>200</td>
<td>200</td>
</tr>
<tr>
<td>Chinese</td>
<td>250</td>
<td>300</td>
<td>150</td>
</tr>
<tr>
<td>Arabic</td>
<td>200</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(K Words)

---

**Related Work**

Pradhan, Xue

**OntoNotes: The 90% Solution**
Parallel Data Quantities: Full OntoNotes

Parallel data with full OntoNotes coverage

- **NW – ECTB – Sinorama**: 154
- **NW – ECTB – Xinhua**: 133
- **BC – Chinese source**: 53
- **BC – English source**: 57
- **K Words**:
  - **English**: 195
  - **Chinese**: 100

Translation Direction

Pradhan, Xue
OntoNotes: The 90% Solution
Parallel Data Quantities: Only Treebank

- P2.5 – Web – Chinese source
- P2.5 – BC – Chinese source
- P2.5 – BN – Chinese source
- P2.5 – NW – Chinese source
- Web – English source
- Web – Chinese source

Data

OntoNotes in a Nutshell

What is OntoNotes?

Tutorial Overview

An Example

Related Work

Pradhan, Xue

OntoNotes: The 90% Solution
Issues with Parallel Data

- **Trade-offs**
  - Translated data may not be predictive of the language as a whole
  - Translated versions of informal genres might end up more text-like

- **Issues**
  - Long lead time involved
    - Data selection and translation
    - Treebanking
    - Propbanking, Word Sense, Coreference
And one of the longest running struggles for international justice reached a milestone today of sorts, when a Scottish court, meeting in the Netherlands, finally officially found someone guilty in the 1988 bombing that brought down Pan Am Flight 103. …

A split decision for Lamen Khalifa Fhimah, acquittal, but Abdel Basset Ali Al-megrahi found guilty as charged. …

The court ruled this senior Libyan intelligence agent planted the bomb that killed 270, mostly Americans, when the plane bound for New York exploded over Lockerbie, Scotland.
The court ruled this senior Libyan intelligence agent planted the bomb that killed 270, mostly Americans, when the plane bound for New York exploded over Lockerbie, Scotland.

NORP (Nationality, Organization, Religious, Political)
GPE
Cardinal
Example (Parse Tree)

```
(S (NP-SBJ (DT this)
  (JJ senior)
  (JJ Libyan)
  (NN intelligence)
  (NN agent))
 (VP (VBD planted)
  (NP (NP (DT the)
       (NN bomb))
  (SBAR (WHNP-1 (WDT that))
    (S (NP-SBJ (-NONE- *T*-1))
     (VP (VBD killed)
      (NP (NP (CD 270))
       (, ,)
       (NP (ADVP (RB mostly))
        (NNPS Americans)))
      (, ,)
    )))
)
...
```
Example (PropBank)

9 planted (PB frame: plant.01)
ARG0 4:1 this senior Libyan intelligence agent
ARG1 10:2 the bomb that *T*-1 killed 270, mostly Americans, when the plane bound * for New York exploded over Lockerbie, Scotland *T*-2

14 killed (PB frame: kill.01)
ARG0 13:0 *T*-1
12:1 that
LINK-SLC 10:1 the bomb
ARG1 15:2 270, mostly Americans
ARGM-TMP 20:2 when the plane bound * for New York exploded over Lockerbie, Scotland *T*-2
Example (Word Sense)

Court-N
1: a sovereign regime and its assemblage
2: assembly that transacts judicial business
3: demarcated area for sports play
4: a room in which judicial proceedings occur
8: respectful deference

Plant-V
1: place into the ground for growing
2: place firmly
3: place secretly, often for later discovery
4: establish, settle

Kill-V
1: cause death, be fatal
2: cause great pain or anguish
3: eliminate
4: thwart
9: drink down
Example (Coreference)

Chain 000-8 (IDENT)
- 0.31-31 someone
- 1.0-0 He
- 5.11-16 Abdel Basset Ali Al-megrahi
- 6.4-8 this senior Libyan intelligence agent
- 12.1-4 Al-megrahi's
- 14.10-10 he

Chain 000-20 (IDENT)
- 1.24-25 the victims
- 4.18-24 the victims of Pan Am Flight 103
- 6.15-18 270, mostly Americans
- 14.7-8 270 people

Chain 000-9 (IDENT)
- 0.34-44 the 1988 bombing that T-3 brought down Pan Am Flight 103
- 6.28-28 exploded
- 20.33-34 this act
- 22.32-33 this crime
The court ruled this senior Libyan intelligence agent planted the bomb that killed 270, mostly Americans, when the plane bound for New York exploded over Lockerbie, Scotland.
Compared to other resources

<table>
<thead>
<tr>
<th>Annotations</th>
<th>WordNet/ SemCor</th>
<th>Salsa</th>
<th>Prague</th>
<th>OntoNotes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Propositions</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Sense Tags ITA &gt; 70-80%</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Sense Tags ITA &gt; 90%</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coref</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>&gt; 1M words</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Genres</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2 + 3*</td>
</tr>
<tr>
<td>Languages</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2* English, 3* English, Chinese, Arabic</td>
</tr>
</tbody>
</table>

*NW, BN, +BC, NG, WebLogs

*English, Chinese, Arabic
Annotation Layers

1. Treebank
2. PropBank
3. Word Sense
4. Ontology
5. Coreference
6. Names
Data Access API

7 Challenges with Multiple Layers of Annotation
8 Architecture
9 Raw Data
10 Database Design
11 Python API Design
12 Data Access
Part I

Annotation Layers
Syntactic Structure

- Phrase Types
- Function Tags
- Traces and Co-indexing

```
S
  └─ TOP
      ├─ NP-TTL-SBJ-1
      │   └─ NNP
      │       └─ "Lighthouse"
      │           └─ NNP
      │               └─ II
      └─ VP
          └─ VBD
              └─ VBN
                  └─ -NONE-
                      └─ IN
                          └─ -NONE-
                              └─ IN
                                  └─ NP-LGS
                                      └─ NNS
                                          └─ IN
                                              └─ NP
                                                  └─ DT
                                                      └─ NN
                                                          └─ CD
```

"Lighthouse II" was painted in oils by the playwright in 1901 -
The Mortgage and equity real estate investment trust last paid a dividend on August 1, 1988.
The Mortgage and equity real estate investment trust last paid a dividend on August 1, 1988.
The Mortgage and equity real estate investment trust last paid a dividend on August 1, 1988.
Adding NP internal structure: NMLs

- Penn Treebank 2 left prenominals flat
- NML constituents fill in that structure
  - Assume a default right-branching structure
  - Specify NMLs where necessary

(NP (DT a)
  (NML (CD 10,000) (NN square) (NN meter))
  (NN visitor)
  (NN center))

(NP (DT this)
  (NML (JJ large) (HYPH -) (NN scale))
  (NML (NN light) (CC and) (NN music))
  (NN show)))))))

PTB2

Right-Branching

With NML
Improving English Treebank Consistency: Hyphenization

- Original treebank did not split any hyphens
- More recent treebanks were not very consistent on which hyphenated tokens were split.
  - This complicates things for parser and parser evaluations
- Trees were revised to split consistently on “most” hyphens
  - Add a GW (goes with) POS tag
    - Covers elements like “co-” in “co-operate”
  - Insert appropriate tree structure over the newly split tokens
  - For any additional layers of existing annotation (including PropBank and Word Sense):
    - Adjust token-based pointers
    - Annotate additional examples in newly-exposed tokens
Chinese Treebanking

- Penn English Treebank approach
  - Phrase structure annotation
  - Emphasis on trade-offs of annotation speed and consistency
  - ITA: 94%

- With enriched structures
  - All structures build on four primitive structures
If among the 100 pieces of news one piece is made up, the reader will also doubt the other 99 pieces.
If among the 100 pieces of news one piece is made up, the reader will also doubt the other 99 pieces.

If among the 100 pieces of news one piece is made up, the reader will also doubt the other 99 pieces.
The Mortgage and equity real estate investment trust last paid a dividend on August 1, 1988.
The Mortgage and equity real estate investment trust last paid a dividend on August 1, 1988.
(Hypothetical) Chinese Treebank

The Mortgage and equity real estate investment trust last paid a dividend on August 1, 1988.
Complementation (left-headed)

Complementation (left-headed)
XP
X YP {ZP}
DP
DeTerminer QP
VP
VV NP
PP P NP

Pradhan, Xue
OntoNotes: The 90% Solution
Adjunction
Coordination

Pradhan, Xue  OntoNotes: The 90% Solution
Propositional Structure

- Tells Who did What to Whom, When, Where, How, etc.
- For both verbs and nouns

Concerns about the pace of the Vienna talks -- which are aimed at the destruction of some 100,000 weapons, as well as major reductions and realignments of troops in central Europe -- also are being registered at the Pentagon.
Predicate frames define the meanings of the numbered arguments.

Concerns about the pace of the Vienna talks -- which are aimed at the destruction of some 100,000 weapons, as well as major reductions and realignments of troops in central Europe -- also are being registered at the Pentagon.

reduce.01 – Make less

ARG0 – Agent
ARG1 – Thing falling
ARG2 – Amount fallen
ARG3 – Starting point
ARG4 – Ending point

- of troops
- major
-
Frame Examples: *expect*, *replace*

**Portfolio managers expect further declines in interest rates**

*expect.01 – Look forward to; anticipate*

ARG0 – Expecter → Portfolio managers
ARG1 – Thing expected → further declines in interest rates

**Continental Air replaced its top executive for the sixth time in as many years**

*replace.01 – substitute*

ARG0 – replacer → Continental Air
ARG1 – old thing → Its top executive
ARG2 – new thing → for the sixth time in as many years
Frame Examples: *increase*

**Net income** increased to $274 million from $130 million

**increase.01** – go up incrementally

- ARG0 – causer of increase
- ARG1 – thing increasing
- ARG2 – amount increased by
- ARG3 – starting point
- ARG4 – end point

Net income → -
Net income → from $130 million
To $274 million
Some word sense distinctions do not change the type of argument that a predicate can take, but some do. Propbank makes only sense distinctions that necessitate a different argument structure or when the argument have different meaning.

- Mary left the room
- If he knew how to handle the finances, I’d leave him lots of money

```
leave.01 – move away from
ARG0 – entity leaving
ARG1 – place left

leave.02 – give
ARG0 – giver
ARG1 – thing given
ARG2 – beneficiary
```
Trends in Argument Numbering

- Arg0 = agent
- Arg1 = direct object/theme/patient
- Arg2 = indirect object/benefactive/instrument/attribute/end state
- Arg3 = start point/benefactive/instrument/attribute
- Arg4 = end point

Consistency for Arg0 and Arg1, but not so much for Arg2, Arg3, ...
Additional tags: ArgMs (arguments or adjuncts?)

- TMP: When?
- LOC: Where at?
- DIR: Where located?
- MNR: How?
- PRP: Why?
- REC: himself, themselves, each other
- PRD: This argument refers to, or modifies another
- ADV: Catch all
Annotation Procedure

**Frame creation**

- Frame creation: Argument definitions, examples, etc.
  (1 person)

**Annotation**

- Automatic tagging (machine)
- Double blind hand correction and frame sense tagging
  (2 people)

  - not ok
  - Results: ok agreement?

  - ok
  - Adjudication: fix remainder
    (1 person)
Chinese PropBank

- Similar in style to English PropBank
  - Predicate-specific numbered labels for core arguments
  - ArgMs for adjunctive arguments
  - Coarse-gained senses

- There are some differences
  - In how split arguments are handled
  - Multi-word expressions are dealt with
If among the 100 pieces of news one piece is made up, the reader will also doubt the other 99 pieces.
Traces and Split Arguments in English PropBank

- **Traces**
  
  [What matters is what advertisers will pay]-1, said *T*-1 Newsweek's chairman

  REL: said
  Arg1: *T*
  Arg0: Newsweek's chairman

- **Split Arguments**

  "What you have to understand," said John [*?*], "is that Philly literally stinks."

  Arg1: [*?*] → ["What you have to understand"] ["is that Philly literally stinks"]
  REL: said
  Arg0: John
中国纺织工业承建*T*的最大项目
Chinese textile industry take on DE largest project
“the largest project that the Chinese textile industry has taken on so far”

ARGM-TMP: 目前为止 “so far”
ARG0: 中国 纺织 工业 “Chinese textile industry”
REL: 承建 “take on”
ARG1: *T* → 最大 项目 “largest project”
Maotai liquor brewing process complex, production cycle long.
“The brewing process of Maotai Liquor is complex, and its production cycle is long.”

REL: 复杂 “complex”
ARG0-PSR: 茅台酒 “Maotai liquor”
ARG0-PSE: 制作工艺 “brewing process”

REL: 长 “long”
ARG0-PSE: 茅台酒 “Maotai Liquor”
ARG0-PSE: 生产周期 “production cycle”
Three main law need accelerate promulgation process.

“The promulgation process of the three main laws need to be accelerated.”

PRED: “accelerate”
ARG1-PSR: “three main laws”
ARG1-PSE: “promulgation process”
West African economy clearly resumed growing.

“West African economy clearly resumed growing”

ARG0: 西非经济 “West African economy”
PRED: 恢复 “resume”
ARGM-ADV: 明显 “clearly”
ARG0-PRD: 增长 “grow”
Reconciling Treebank and PropBank

- We found several mismatches between syntax and propositions
  - Sometimes PropBank was right
  - Somethings Treebank was right
- Ambiguities were resolved (PP-attachment)
- Guidelines were modified to bring the two in sync
  - Modified list of verbs that take small-clauses and sentential complements (eg. keep their markets active)
  - A different approach to annotation of empty categories
- Now each argument points to a single node in the tree
  - Secondary connections are made using Treebank trace chains
  - Almost no discontinuous arguments
  - Non-trace connections are explicitly identified as LINK-SLC and LINK-PCR
WordNet, OntoNotes and PropBank senses for develop-v

- Create
- Come about
- Alter by chemical means
- Bring into existence
- Superimpose
- Further grow
- Come about
Sense Annotation Procedure

Word

Sense creation:
definitions, examples, etc.
(1 person)

Annotation

Pre-annotation: 50 instances
(2 people)

not ok

Results: ok agreement?

ok

Full annotation: all instances
(2 people)

not ok

Results: ok agreement?

ok

Adjudication: fix remainder
(1 person)
Word Sense and Ontology

- Meaning of nouns and verbs are specified using a catalog of possible senses
- All the senses are annotatable at ~90% ITA
- Ontology links (currently being added) capture similarities between related senses of different words

Concerns about the pace of the Vienna talks -- which are aimed at the destruction of some 100,000 weapons, as well as major reductions and realignments of troops in central Europe -- also are being registered at the Pentagon.

**aim**
1. Point or direct object, weapon, at something ...
2. Wish, purpose or intend to achieve something

**register**
1. Enter into an official record
2. Be aware of, enter into someone’s consciousness
3. Indicate a measurement
4. Show in one’s face
Ontologizing

Pradhan, Xue  OntoNotes: The 90% Solution
Ontologizing

Sense Pooling Procedure

Collect synonyms
(1 person)

Create sense pools
(1 person)

Validate sense pools
(2 people)

Results: ok agreement?
ok
not ok

Taxonomize pools
(1 person)

Store results
in ontology
Snapshot of the Upper Model
Snapshot of the Ontology
Ontology Structure

- Upper Model
  - 150 concepts
- Sense Pools
  - 3000 Sense Pools
- Links
  - Subtype
  - Related
Coreference

- Identifies different mentions of the same entity within a document – especially links definite, referring noun phrases, and pronouns to their antecedents
- Two types tagged - Identity (IDENT) and Attributive (APPOS)

Concerns about the pace of the Vienna talks – which are aimed at the destruction of some 100,000 weapons, as well as major reductions and realignments of troops in central Europe – also are being registered at the Pentagon.
Salient points

- All types of entities, and even events (marked by verbs) are coreferenced.
- Barring few exceptions (2%) coreference links are typically restricted to nodes in the syntax trees.
- Name, nominal and pronoun mentions are coreferenced.
- In pro-drop languages like Chinese and Arabic, the “*” or “*pro*” in the tree are tagged with coreference.
- Generic, underspecified mentions are not coreferenced.
- Singleton mentions are not coreferenced.
- Copulas are not coreferenced with each other.
- Only intra-document coreference is marked – When document lengths were prohibitive, they were broken down into parts and individual part independently annotated.
Coreference Annotation Examples

- IDENT
  - [Elco Industries Inc]_x said [it]_x expects net income to fall below a recent estimate of $1.65 a share. [The Rockford, Ill. maker of fasteners]_x also said that [it]_x expects to post sales in the current fiscal year that are “slightly above” fiscal 1989 sales of $155 million.
  - Sales of passenger cars [grew]_x 22%. [The strong growth]_x followed year-to-year increases.

- APPOS
  - [[The PhacoFlex intraocular lens]_HEAD, [the first foldable silicone lens available for cataract surgery]_ATTRIB]_x
Special Challenges in the Broadcast Conversation Data

- **Disfluency Effects**
  
  Former Iraqi war combat veteran I guess 0 <disfluency> he ’s a -- --
  
  </disfluency> he is a present veteran Paul Hackett

- **Ambiguity in speaker turn labels**
  
  <Firefighter_A> It began as <disfluency> an- </disfluency> any other day you know
  
  *PRO* <uncertain> just uh </uncertain> doing eh normal checks .
  
  <Firefighter_B> At nine o’clock we started our shift .
  
  <Firefighter_A> And so the bells went .
  
  <Firefighter_B> It was about a minute past nine when we got the shout for uh
  
  <uncertain> smoke issuing </uncertain> in Allgate tube station *T*-1 .
  
  <Andrew_Carey> The explosion at Allgate was the first of the four bombs 0 *T*-1 to go off on July the seventh at eight fifty in the morning .
  
  <Andrew_Carey> But Paul Kelly Steve Sodbury and Mel Anderson of <uncertain>
  
  Shadwell </uncertain> Firestation ’s blue watch had no idea what *T*-1 had happened as they got into the fire engine *PRO*-2 to answer the call .
Names
Types of Names (I)

- Person – People, including fictional
- NORP – Nationalities, or religious or political groups
- Facility – Buildings, airports, highways, bridges, etc.
- Organization – Companies, agencies, institutions, etc.
- GPE – Countries, cities, states, etc.
- Location – Non-GPE locations, mountain ranges, bodies of water
- Product – Vehicles, weapons, foods, etc.
- Event – Named hurricanes, battles, wars, etc.
- Work of Art – Titles of books, songs, etc.
Types of Names (II)

- Law – Named documents made into law
- Language – Any named language
- Date – Absolute or relative dates or periods
- Time – Times smaller than a day
- Percent – Percentage
- Money – Monetary values – including unit
- Quantity – Measurements as of weight and distance
- Ordinal – “First”, “Second”, etc.
- Cardinal – Numerals that do not fall under another type
Part II

Integrated representation
Interpreting Tree pointers in Propositions

Pradhan, Xue
OntoNotes: The 90% Solution
Interpreting Argument meaning and constraints

wsj_0037.mrg 67 5 gold set.02 ----- 0:2-ARG0 5:0-rel 6:1-ARG1 10:2-ARGM-TMP
wsj_0037.mrg 69 21 gold exchange.01 ----- 17:2-ARG0 21:0-rel 22:1-ARG1 23:1-ARGM-TMP
wsj_0037.mrg 69 35 gold say.01 ----- 31:1-ARG0 35:0-rel 0:2*37:0-ARG1

<!DOCTYPE frameset SYSTEM "frameset.dtd">
<frameset>
  <predicate lemma="paint">
    <note>
      Frames file for 'paint' based on sentences in wsj and automatic expansion via verbnet.
    </note>
    <roleset id="paint.01" name="put paint on a surface" vncls="25.1">
      <roles>
        <role descr="agent, painter" n="0">
          <vnrole vncls="25.1" vntheta="Agent"/>
        </role>
        <role descr="surface" n="1">
          <vnrole vncls="25.1" vntheta="Destination"/>
        </role>
        <role descr="explicit mention of paint" n="2">
          <vnrole vncls="25.1" vntheta="Theme"/>
        </role>
      </roles>
    </roleset>
  </predicate>
</frameset>
Interpreting Sense Numbers and their mappings

*PRO* Judging from the Americana in Haruki Murakami’s “A Wild Sheep Chase” (Kodansha, 320 pages, $18.95 “U”), baby boomers on both sides of the Pacific have a lot in common.

```
<?xml version="1.0" ?>
<!DOCTYPE inventory SYSTEM "inventory.dtd">
<inventory lemma="judge-v">
  <sense group="1" n="1" name="act as an official judge">
    <examples>She was asked to judge the fancy-dress competition.</examples>
    <mappings>
      <wn version="2.1">1,5</wn>
      <pb>judge.01</pb>
    </mappings>
  </sense>
  <sense group="1" n="2" name="form an opinion, or conclusion">
    <examples>They quickly judged him unfit to join the team.</examples>
    <mappings>
      <wn version="2.1">2,3,4</wn>
      <pb>judge.01</pb>
    </mappings>
  </sense>
</inventory>
```
Challenge with Multiple Layers of Annotation

- Not previously available
  - A number of these layers have not been available in significant quantity before:
    - Word Sense
    - Coreference
- Not previously integrated
- Not previously completely consistent
- Not previously easily accessible
  - Raw text format
- Not user friendly
Challenges with Multiple Layers of Annotation

- **Not** previously available
  - A number of these layers have not been available in significant quantity before:
    - Word Sense
    - Coreference

- **Not** previously integrated
- **Not** previously completely consistent
- **Not** previously easily accessible
  - Raw text format
- **Not** user friendly
Challenge with Multiple Layers of Annotation

- **Not** previously available
  - A number of these layers have not been available in significant quantity before:
    - Word Sense
    - Coreference

- **Not** previously integrated
  - Not previously completely consistent
  - Not previously easily accessible
    - Raw text format
  - Not user friendly

Pradhan, Xue
OntoNotes: The 90% Solution
Challenge with Multiple Layers of Annotation

- Not previously available
  - A number of these layers have not been available in significant quantity before:
    - Word Sense
    - Coreference
- Not previously integrated
- Not previously completely consistent
  - Not previously easily accessible
    - Raw text format
  - Not user friendly
Challenge with Multiple Layers of Annotation

- **Not** previously available
  - A number of these layers have not been available in significant quantity before:
    - Word Sense
    - Coreference

- **Not** previously integrated
- **Not** previously completely consistent
- **Not** previously easily accessible
  - Raw text format

- **Not** user friendly

Pradhan, Xue

OntoNotes: The 90% Solution
Challenge with Multiple Layers of Annotation

- **Not** previously available
  - A number of these layers have not been available in significant quantity before:
    - Word Sense
    - Coreference

- **Not** previously integrated
- **Not** previously completely consistent
- **Not** previously easily accessible
  - Raw text format
- **Not** user friendly
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra- and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra-and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra-and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra-and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra-and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
    - Integrate information at the lowest level of granularity
    - Robust to superficial changes in representations
- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra-and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra- and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
A Solution: Unified Representation

- Provide a bare-bones representation independent of the individual semantics that can
  - Efficiently capture intra-and inter-layer semantics
  - Maintain component independence
  - Provide mechanism for flexible integration
  - Integrate information at the lowest level of granularity
  - Robust to superficial changes in representations

- A Relational Database + Object Oriented API
Modes of Data Access

- SQL queries can extract examples based on multiple layers or define new views
- Python Object-Oriented API allows for programmatic access to tables and queries
- And, the raw text files as well
Modes of Data Access

- SQL queries can extract examples based on multiple layers or define new views.
- Python Object-Oriented API allows for programmatic access to tables and queries.
- And, the raw text files as well.
Modes of Data Access

- SQL queries can extract examples based on multiple layers or define new views
- Python Object-Oriented API allows for programmatic access to tables and queries
- And, the raw text files as well
Advantages of Integrated Representation

- Each layer translates into a common representation
- Clean consistent Layers
- Well defined relationships – The Database scheme defines the merged structure efficiently
- Original representations available as pre-defined views, eg. Treebank, PropBank, etc.
- SQL queries can extract examples based on multiple layers or define new views
- Python object-oriented API allows for programmatic access to tables and queries
Advantages of Integrated Representation

- Each layer translates into a common representation
- Clean consistent Layers
  - Well defined relationships – The Database scheme defines the merged structure efficiently
  - Original representations available as pre-defined views, eg. Treebank, PropBank, etc.
  - SQL queries can extract examples based on multiple layers or define new views
  - Python object-oriented API allows for programmatic access to tables and queries
Advantages of Integrated Representation

- Each layer translates into a common representation
- Clean consistent Layers
- Well defined relationships – The Database scheme defines the merged structure efficiently
- Original representations available as pre-defined views, eg. Treebank, PropBank, etc.
- SQL queries can extract examples based on multiple layers or define new views
- Python object-oriented API allows for programmatic access to tables and queries
Advantages of Integrated Representation

- Each layer translates into a common representation
- Clean consistent Layers
- Well defined relationships – The Database scheme defines the merged structure efficiently
- Original representations available as pre-defined views, eg. Treebank, PropBank, etc.
- SQL queries can extract examples based on multiple layers or define new views
- Python object-oriented API allows for programmatic access to tables and queries
Advantages of Integrated Representation

- Each layer translates into a common representation

- Clean consistent Layers

- Well defined relationships – The Database scheme defines the merged structure efficiently

- Original representations available as pre-defined views, eg. Treebank, PropBank, etc.

- SQL queries can extract examples based on multiple layers or define new views

- Python object-oriented API allows for programmatic access to tables and queries
Advantages of Integrated Representation

- Each layer translates into a common representation
- Clean consistent Layers
- Well defined relationships – The Database scheme defines the merged structure efficiently
- Original representations available as pre-defined views, eg. Treebank, PropBank, etc.
- SQL queries can extract examples based on multiple layers or define new views
- Python object-oriented API allows for programmatic access to tables and queries
Data Lifecycle

Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Advantages
Data Lifecycle

Pradhan, Xue
OntoNotes: The 90% Solution
Organization of the OntoNotes data

.../data/\texttt{<lang>}/annotations/\texttt{<genre>}/\texttt{<source>}/\texttt{<section>}/\texttt{<filename>}.\texttt{<extension>}
.../data/\texttt{<lang>}/metadata/\texttt{<inventory-type>}/\texttt{<filename>}.xml

\texttt{<inventory-type>} ::= ("frames" | "sense-inventories")
\texttt{<extension>} ::= ("parse" | "prop" | "sense" | "coref" | "name" | "parallel" | "speaker")
Entity Relationship Diagram (I)
Entity Relationship Diagram (II)
The corpus tables collectively manage information about the corpus – specifically the subcorpora, documents, files, etc.
The treebank tables manage the syntactic tree information. Tokens form the lowest level of granularity in OntoNotes.
The proposition tables manage the propositions. The `argument_node` forms a composite table to manage many-to-many argument/node relationships.
The sense tables contain the lemma and sense number representing its sense

Multiple composite tables are used to map WordNet sense, OntoNotes sense and Frame senses to each other
**Coreference Tables**

- The `coreference_chain` and `coreference_link` tables store the respective pointers.

Pradhan, Xue

OntoNotes: The 90% Solution
The name_entity and name_type tables represent the names in the corpus.
Module Organization

- Corpora
  - Tree
  - Proposition
  - Name
  - Sense
  - Coreference
  - Ontology
  - Parallel
  - Speaker

- Common
  - Util
  - Log
Object Composition

Pradhan, Xue
OntoNotes: The 90% Solution
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Python Modules
Database/API Correspondence

on.corpora.tree

Pradhan, Xue

OntoNotes: The 90% Solution
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Python Modules
Database/API Correspondence

On.corpora.proposition

frame_set

+ lemma
+ subcorpus
+ __init__()
+ __repr__()
+ write_to_db(cursor)

predicate

+ id
+ enc_predicate
+ index
+ lemma
+ pb_sense_num
+ proposition_id
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ __repr__()
+ write_to_db(cursor)
+ propositions 1..*

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()

predicate_part

+ id
+ type
+ __init__()
+ __repr__()
+ write_to_db(cursor)

argument

+ id
+ encoded_argument_analogue
+ argument_analogue_index
+ argument_index
+ argument_analogue
+ __init__()
+ __repr__()

argument_analogue

+ id
+ lemma
+ num
+ type_hash: variant
+ __init__()
+ __repr__()
+ write_to_db(cursor)

predicate_type

+ id
+ type_hash
+ type
+ __init__()
+ __repr__()
+ write_to_db(cursor)

argument_type

+ id
+ type
+ __init__()
+ __repr__()
+ write_to_db(cursor)

predicate_part

+ id
+ type
+ __init__()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)
+ propositions 1..*

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()

predicate

+ id
+ enc_predicate
+ index
+ lemma
+ pb_sense_num
+ proposition_id
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ __repr__()
+ write_to_db(cursor)

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()

predicate

+ id
+ enc_predicate
+ index
+ lemma
+ pb_sense_num
+ proposition_id
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ __repr__()
+ write_to_db(cursor)

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()

predicate

+ id
+ enc_predicate
+ index
+ lemma
+ pb_sense_num
+ proposition_id
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ __repr__()
+ write_to_db(cursor)

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()

predicate

+ id
+ enc_predicate
+ index
+ lemma
+ pb_sense_num
+ proposition_id
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ __repr__()
+ write_to_db(cursor)

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()

predicate

+ id
+ enc_predicate
+ index
+ lemma
+ pb_sense_num
+ proposition_id
+ __init__()
+ get_primary_predicate()
+ __repr__()
+ write_to_db(cursor)

proposition

+ id
+ document_id
+ corpus_id
+ sentence_index
+ predicate_type
+ lemma_and_sense
+ lemma
+ pb_sense_num
+ __init__()
+ __repr__()
+ write_to_db(cursor)

proposition_bank

+ id
+ subcorpus
+ lemma_hash
+ tag
+ __init__()
+ enrich_treebank(treebank)
+ write_to_db(cursor)
+ from_db(cursor)
+ dump_view()
Every bank has a `enrich_treebank` method which takes a `treebank` object and aligns itself to the trees.

Almost every object has a `from_db` and `write_to_db` method which can create itself from the database, or serialize itself to the database.

The SQL statements for reading/writing to DB are class attributes of most classes.
<table>
<thead>
<tr>
<th>Bank Name</th>
<th>Database Table</th>
<th>Python Module</th>
<th>Extention</th>
</tr>
</thead>
<tbody>
<tr>
<td>tree</td>
<td>tree</td>
<td>on.corpora.tree</td>
<td>.parse</td>
</tr>
<tr>
<td>sense</td>
<td>on_sense</td>
<td>on.corpora.sense</td>
<td>.sense</td>
</tr>
<tr>
<td>proposition</td>
<td>argument, predicate</td>
<td>on.corpora.proposition</td>
<td>.prop</td>
</tr>
<tr>
<td>coreference</td>
<td>coreference_link</td>
<td>on.corpora.coreference</td>
<td>.coref</td>
</tr>
<tr>
<td>name</td>
<td>name_entity</td>
<td>on.corpora.name</td>
<td>.name</td>
</tr>
<tr>
<td>speaker</td>
<td>speaker_sentence</td>
<td>on.corpora.speaker</td>
<td>.speaker</td>
</tr>
<tr>
<td>parallel</td>
<td>parallel_sentence, parallel_document</td>
<td>on.corpora.parallel</td>
<td>.parallel</td>
</tr>
</tbody>
</table>
## Challenges with Multiple Layers of Annotation

### Architecture
- Raw Data
- Database Design
- Python API Design
- Data Access

### Python Modules
- Database/API Correspondence

### Treebank

<table>
<thead>
<tr>
<th>Database Tables</th>
<th>Python Objects</th>
<th>File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>treebank</td>
<td>treebank</td>
<td>All <code>.parse</code> files for a <code>on.corpora.subcorpus</code></td>
</tr>
<tr>
<td>None</td>
<td>tree_document</td>
<td>A <code>.parse</code> file</td>
</tr>
<tr>
<td>tree</td>
<td>tree</td>
<td>An S-expression in a <code>.parse</code> file</td>
</tr>
<tr>
<td>syntactic_link</td>
<td>syntactic_link</td>
<td>The numbers after <code>-' and </code>=' in trees</td>
</tr>
<tr>
<td>lemma</td>
<td>lemma</td>
<td><code>.lemma</code> files (arabic only)</td>
</tr>
</tbody>
</table>
### PropBank

<table>
<thead>
<tr>
<th><strong>Database Tables</strong></th>
<th><strong>Python Objects</strong></th>
<th><strong>File Elements</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>proposition_bank</code></td>
<td><code>proposition_bank</code></td>
<td>All <code>.prop</code> files in an on.corpora.subcorpus</td>
</tr>
<tr>
<td>None</td>
<td><code>proposition_document</code></td>
<td>A single <code>.prop</code> file</td>
</tr>
<tr>
<td><code>proposition</code></td>
<td><code>proposition</code></td>
<td>A line in a <code>.prop</code> file, with everything after the <code>-----</code> an “argument field”</td>
</tr>
<tr>
<td>None</td>
<td><code>predicate_analogue</code></td>
<td>REL argument fields (should only be one)</td>
</tr>
<tr>
<td>None</td>
<td><code>argument_analogue</code></td>
<td>ARG argument fields</td>
</tr>
<tr>
<td>None</td>
<td><code>link_analogue</code></td>
<td>LINK argument fields</td>
</tr>
<tr>
<td><code>predicate</code></td>
<td><code>predicate</code></td>
<td>Asterisk-separated components of a predicate_analogue. Each part is coreferential.</td>
</tr>
<tr>
<td><code>argument</code></td>
<td><code>argument</code></td>
<td>Asterisk-separated components of an argument_analogue. Each part is coreferential.</td>
</tr>
<tr>
<td><code>proposition_link</code></td>
<td><code>link</code></td>
<td>Asterisk-separated components of a link_analogue. Each part is coreferential.</td>
</tr>
<tr>
<td><code>predicate_node</code></td>
<td><code>predicate_node</code></td>
<td>Comma-separated components of predicates. The parts together make up the predicate.</td>
</tr>
<tr>
<td><code>argument_node</code></td>
<td><code>argument_node</code></td>
<td>Comma-separated components of arguments. The parts together make up the argument.</td>
</tr>
<tr>
<td><code>link_node</code></td>
<td><code>link_node</code></td>
<td>Comma-separated components of links. The parts together make up the link.</td>
</tr>
<tr>
<td>None</td>
<td><code>frame_set</code></td>
<td>An xml frame file (FF)</td>
</tr>
<tr>
<td><code>pb_sense_type</code></td>
<td><code>on.corpora.sense.pb_sense_type</code></td>
<td>Field six of a prop line and a FF’s <code>frameset/predicate/roleset</code> element’s <code>id</code> attribute</td>
</tr>
<tr>
<td><code>pb_sense_type_argument_type</code></td>
<td><code>argument_composition</code></td>
<td>For a FF’s <code>frameset/predicate</code> element, a mapping between...</td>
</tr>
</tbody>
</table>
### Word Sense

<table>
<thead>
<tr>
<th>Database Tables</th>
<th>Python Objects</th>
<th>File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>sense_bank</td>
<td>sense_bank</td>
<td>All .sense files in a on.corpora.subcorpus</td>
</tr>
<tr>
<td>None</td>
<td>sense_tagged_document</td>
<td>A single .sense file</td>
</tr>
<tr>
<td>on_sense</td>
<td>on_sense</td>
<td>A line in a .sense file</td>
</tr>
<tr>
<td>None</td>
<td>sense_inventory</td>
<td>A sense inventory xml file (SI)</td>
</tr>
<tr>
<td>on_sense_type</td>
<td>on_sense_type</td>
<td>Fields four and six of a sense line and the inventory/sense element of a SI</td>
</tr>
<tr>
<td>on_sense_lemma_type</td>
<td>on_sense_lemma_type</td>
<td>The inventory/ita element of a SI</td>
</tr>
<tr>
<td>wm_sense_type</td>
<td>wm_sense_type</td>
<td>The inventory/sense/mappings/wm element of a SI</td>
</tr>
<tr>
<td>pb_sense_type</td>
<td>pb_sense_type</td>
<td>The inventory/sense/mappings/pb element of a SI</td>
</tr>
<tr>
<td>tree</td>
<td>on.corpora.tree.tree</td>
<td>The first three fields of a sense line</td>
</tr>
</tbody>
</table>
## Challenges with Multiple Layers of Annotation

### Architecture
- Raw Data
- Database Design
- Python API Design
- Data Access

### Coreference

<table>
<thead>
<tr>
<th>Database Tables</th>
<th>Python Objects</th>
<th>File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>coreference_bank</td>
<td>coreference_bank</td>
<td>All <code>.coref</code> files in an <code>on.corpora.subcorpus</code></td>
</tr>
<tr>
<td>None</td>
<td>coreference_document</td>
<td>A <code>.coref</code> file (a <code>DOC</code> span)</td>
</tr>
<tr>
<td>tree.coreference_section</td>
<td>on.corpora.tree.tree.coref_section</td>
<td>An annotation section of a <code>.coref</code> file (a <code>TEXT</code> span)</td>
</tr>
<tr>
<td>tree</td>
<td>on.corpora.tree.tree</td>
<td>A line in a <code>.coref</code> file</td>
</tr>
<tr>
<td>coreference_chain</td>
<td>coreference_chain</td>
<td>All <code>COREF</code> spans with a given <code>ID</code></td>
</tr>
<tr>
<td>coreference_chain.type</td>
<td>coreference_chain.type</td>
<td>The <code>TYPE</code> field of a coreference link (the same for all links in a chain)</td>
</tr>
<tr>
<td>coreference_link</td>
<td>coreference_link</td>
<td>A single <code>COREF</code> span</td>
</tr>
<tr>
<td>coreference_link.type</td>
<td>coreference_link.type</td>
<td>The <code>SUBTYPE</code> field of a coreference link</td>
</tr>
</tbody>
</table>
### Challenges with Multiple Layers of Annotation Architecture

- Raw Data
- Database Design
- Python API Design
- Data Access

### Python Modules
- Database/API Correspondence

## Name

### DB Tables ↔ Python Objects ↔ File Elements

<table>
<thead>
<tr>
<th>Database Tables</th>
<th>Python Objects</th>
<th>File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>name_bank</td>
<td>name_bank</td>
<td>All <code>.name files</code> in an <code>on.corpora.subcorpus</code></td>
</tr>
<tr>
<td>None</td>
<td>name_tagged_document</td>
<td>A <code>.name file</code></td>
</tr>
<tr>
<td>tree</td>
<td>on.corpora.tree.tree</td>
<td>A line in a <code>.name file</code></td>
</tr>
<tr>
<td>name_entity</td>
<td>name_entity</td>
<td>A single <code>ENAMEX</code>, <code>TIMEX</code>, or <code>NUMEX</code> span</td>
</tr>
<tr>
<td>None</td>
<td>name_entity_set</td>
<td>All <code>name_entity</code> instances for one <code>on.corpora.tree.tree</code></td>
</tr>
</tbody>
</table>
### Inventories

<table>
<thead>
<tr>
<th>Database Tables</th>
<th>Python Objects</th>
<th>File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>sense_bank</td>
<td>sense_bank</td>
<td>All <code>.sense</code> files in a <code>on.corpora.subcorpus</code></td>
</tr>
<tr>
<td>None</td>
<td>sense_tagged_document</td>
<td>A single <code>.sense</code> file</td>
</tr>
<tr>
<td>on_sense</td>
<td>on_sense</td>
<td>A line in a <code>.sense</code> file</td>
</tr>
<tr>
<td>None</td>
<td>sense_inventory</td>
<td>A sense inventory xml file (SI)</td>
</tr>
<tr>
<td>on_sense_type</td>
<td>on_sense_type</td>
<td>Fields four and six of a sense line and the <code>inventory/sense</code> element of a SI</td>
</tr>
<tr>
<td>on_sense_lemma_type</td>
<td>on_sense_lemma_type</td>
<td>The <code>inventory/ita</code> element of a SI</td>
</tr>
<tr>
<td>wn_sense_type</td>
<td>wn_sense_type</td>
<td>The <code>inventory/sense/mappings/wn</code> element of a SI</td>
</tr>
<tr>
<td>pb_sense_type</td>
<td>pb_sense_type</td>
<td>The <code>inventory/sense/mappings/pb</code> element of a SI</td>
</tr>
<tr>
<td>tree</td>
<td>on.corpora.tree.tree</td>
<td>The first three fields of a sense line</td>
</tr>
<tr>
<td>Database Tables</td>
<td>Python Objects</td>
<td>File Elements</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>None</td>
<td>parallel_bank</td>
<td>All .parallel files in an on.corpora.subcorpus</td>
</tr>
<tr>
<td>parallel_document</td>
<td>parallel_document</td>
<td>The second line (original/translation line) in a .parallel file</td>
</tr>
<tr>
<td>parallel_sentence</td>
<td>parallel_sentence</td>
<td>All lines in a .parallel file after the first two (map lines)</td>
</tr>
</tbody>
</table>
### Speaker

<table>
<thead>
<tr>
<th>Database Tables</th>
<th>Python Objects</th>
<th>File Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>speaker_bank</td>
<td>All .speaker files in an on.corpora.subcorpus</td>
</tr>
<tr>
<td>None</td>
<td>speaker_document</td>
<td>A .speaker file</td>
</tr>
<tr>
<td>speaker_sentence</td>
<td>speaker_sentence</td>
<td>A line in a .speaker file</td>
</tr>
</tbody>
</table>
### Configuration File

#### Sections of the Configuration

- **[corpus]**
  - data_in: [</path/to/data>]
  - load: (<lang>-<genre> | <lang>-<genre>-<source>)+
  - prefix: (<prefix>)*
  - suffix: (<suffix>)*
  - granularity: <granularity>
  - banks: (<bank>)+
  - ignore-inventories: (<inventory>)*

- **[db]**
  - db: <ontonotes-database-name>
  - server: <your-mysql-server-address>
  - db-user: <your-mysql-username>

<lang> ::= ("english" | "chinese" | "arabic")
<genre> ::= ("nw" | "bn" | "mz" | "bc")
<source> ::= ("wsj" | "cnn" | "msnbc" | "xinhua" | ...)
<bank> ::= ("parse" | "prop" | "sense" | "coref" | "name" | "parallel" | "speaker")
<inventory> ::= ("senses" | "frames")
<granularity> ::= ("file" | "source" | "genre")
<prefix> ::= <digit>+
<suffix> ::= <digit>+
A Sample Configuration

[corpus]

Pradhan, Xue

OntoNotes: The 90% Solution
A Sample Configuration

[corpus]
data_in : /corpora/ontonotes/v3/data
A Sample Configuration

[corpus]
data_in : /corpora/ontonotes/v3/data
load : english-nw-wsj chinese-bc
## A Sample Configuration

```yaml
[corpus]
data_in : /corpora/ontonotes/v3/data
load : english-nw-wsj chinese-bc
prefix : 02 03
```
A Sample Configuration

```yaml
[corpus]
data_in : /corpora/ontonotes/v3/data
load : english-nw-wsj chinese-bc
prefix : 02 03
suffix :
```

A Sample Configuration

```ini
[corpus]
data_in : /corpora/ontonotes/v3/data
load : english-nw-wsj chinese-bc
prefix : 02 03
suffix : 
granularity : file
```
A Sample Configuration

[corpus]
data_in : /corpora/ontonotes/v3/data
load : english-nw-wsj chinese-bc
prefix : 02 03
suffix :
granularity : file
banks : parse prop sense
A Sample Configuration

[corpus]
data_in : /corpora/ontonotes/v3/data
load : english-nw-wsj chinese-bc
prefix : 02 03
suffix :
granularity : file
banks : parse prop sense
ignore-inventories : senses frames

[db]
A Sample Configuration

```
[corpus]
data_in      : /corpora/ontonotes/v3/data
load         : english-nw-wsj chinese-bc
prefix       : 02 03
suffix       :
granularity  : file
banks        : parse prop sense
ignore-inventories : senses frames

[db]
db           : ontonotes_v3
```

Pradhan, Xue
A Sample Configuration

[corpus]
data_in    : /corpora/ontonotes/v3/data
load       : english-nw-wsj chinese-bc
prefix     : 02 03
suffix     :
granularity: file
banks      : parse prop sense
ignore-inventories: senses frames

[db]
 db       : ontonotes_v3
 server   : ontonotes.bbn.com
A Sample Configuration

[corpus]
data_in : /corpora/ontonotes/v3/data
load    : english-nw-wsj chinese-bc
prefix  : 02 03
suffix  :
granularity : file
banks    : parse prop sense
ignore-inventories : senses frames

[db]
db      : ontonotes_v3
server  : ontonotes.bbn.com
db-user : ontonotes
Configuration File

```
[corpus]
data_in : data
load : english-nw-wsj
granularity : source
banks : parse coref sense name prop parallel speaker
ignore-inventories: senses frames
```
Reading the Configuration

In [1]: import on

In [2]: import on.common.util

In [3]: c = on.common.util.load_config("config.example")

In [4]: c

Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>

In [5]: c.sections()

Out[5]: ['corpus']

In [7]: c["corpus", "banks"]

Out[7]: 'parse coref sense name parallel prop speaker'

Pradhan, Xue

OntoNotes: The 90% Solution
In [1]: import on

In [2]: import on.common.util

In [3]: c = on.common.util.load_config("config.example")

In [4]: c
Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>

In [5]: c.sections()
Out[5]: ['corpus']

In [7]: c["corpus", "banks"]
Out[7]: 'parse coref sense name parallel prop speaker'
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Reading the Configuration

In [1]: import on
In [2]: import on.common.util
Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
In [4]: c
Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
In [4]: c
Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
In [4]: c
Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>

In [5]: c.sections()
Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
In [4]: c
Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>

In [5]: c.sections()
Out[5]: ['corpus']
Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
In [4]: c
Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>

In [5]: c.sections()
Out[5]: ['corpus']

In [7]: c["corpus", "banks"]
Reading the Configuration

In [1]: import on
In [2]: import on.common.util
In [3]: c = on.common.util.load_config("config.example")
In [4]: c
Out[4]: <on.common.util.FancyConfigParser instance at 0x82c4c4c>

In [5]: c.sections()
Out[5]: ['corpus']

In [7]: c["corpus", "banks"]
Out[7]: ‘parse coref sense name parallel prop speaker’
Creating the ontonotes Object

In [9]:
   o = on.ontonotes()

Loading english nw wsj
....................
found 4 files in the subcorpus all@wsj@nw@en@on

In [10]: o
Out[10]:
ontonotes instance, id=on, subcorpora:
   [0] : all@wsj@nw@en@on

Pradhan, Xue
OntoNotes: The 90% Solution
Creating the `ontonotes` Object

```
In [9]: o = on.ontonotes(c)
```
Creating the ontonotes Object

In [9]: o = on.ontonotes(c)
Loading english nw wsj
.................
found 4 files in the subcorpus all@wsj@nw@en@on
Creating the ontonotes Object

In [9]: o = on.ontonotes(c)
Loading english nw wsj
.................
found 4 files in the subcorpus all@wsj@nw@en@on

In [10]: o
Creating the ontonotes Object

In [9]: o = on.ontonotes(c)
Loading english nw wsj

....................
found 4 files in the subcorpus all@wsj@nw@en@on

In [10]: o
Out[10]:
tononotes instance, id=on, subcorpora:
[0] : all@wsj@nw@en@on
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue

OntoNotes: The 90% Solution
In [6]: c["corpus", "granularity"] = "file"
In [6]: c["corpus", "granularity"] = "file"

In [8]: o = on.ononotes(c)
In [6]: c["corpus", "granularity"] = "file"

In [8]: o = on.ontonotes(c)
Loading english nw wsj
.....
found 1 file in the subcorpus 0089@wsj@nw@en@on
.....
found 1 file in the subcorpus 0020@wsj@nw@en@on
.....
found 1 file in the subcorpus 0049@wsj@nw@en@on
.....
found 1 file in the subcorpus 0037@wsj@nw@en@on
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [6]: c["corpus", "granularity"] = "file"

In [8]: o = on.ontonotes(c)

Loading english nw wsj

.....
found 1 file in the subcorpus 0089@wsj@nw@en@on
.....
found 1 file in the subcorpus 0020@wsj@nw@en@on
.....
found 1 file in the subcorpus 0049@wsj@nw@en@on
.....
found 1 file in the subcorpus 0037@wsj@nw@en@on

In [14]: o
In [6]: c["corpus", "granularity"] = "file"

In [8]: o = on.ononotes(c)
Loading english nw wsj
.....
found 1 file in the subcorpus 0089@wsj@nw@en@on
.....
found 1 file in the subcorpus 0020@wsj@nw@en@on
.....
found 1 file in the subcorpus 0049@wsj@nw@en@on
.....
found 1 file in the subcorpus 0037@wsj@nw@en@on

In [14]: o
Out[14]:
ononotes instance, id=on, subcorpora:
[0] : 0089@wsj@nw@en@on
[1] : 0020@wsj@nw@en@on
[2] : 0049@wsj@nw@en@on
[3] : 0037@wsj@nw@en@on
Loading the banks
Loading the banks

```
In [11]: s = o[0]
```
Loading the banks

In [11]: s = o[0]
Loading banks for all@wsj@nw@en@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] ........ 233 trees in the treebank
reading the coreference bank [coref] ........
Enriching parse with coref ...
Loading the banks

In [11]: s = o[0]
Loading banks for all@wsj@nw@en@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] ....... 233 trees in the treebank
reading the coreference bank [coref] .......
Enriching parse with coref ...

reading the sense bank [sense] ........
Enriching parse with sense ...
....
Loading the banks

In [11]: s = o[0]
Loading banks for all@wsj@nw@en@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] ....... 233 trees in the treebank
reading the coreference bank [coref] .......
Enriching parse with coref ...

reading the sense bank [sense] .......
Enriching parse with sense ...
....
reading the name bank [name].......Enriching parse with name ...
....
reading the parallel bank [parallel] ...keys: ['parse', 'prop', 'coref', 'name', 'sense']
Pradhan, Xue

In [11]: s = o[0]
Loading banks for all@wsj@nw@en@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] ........ 233 trees in the treebank
reading the coreference bank [coref] ........
Enriching parse with coref ...

reading the sense bank [sense] ........
Enriching parse with sense ...
....
reading the name bank [name]....... 
Enriching parse with name ...
....
reading the parallel bank [parallel] ...keys: ['parse', 'prop', 'coref', 'name', 'sense']

reading the proposition bank [prop] ........
Enriching parse with prop ...
....
reading the speaker bank [speaker] ...keys: ['parse', 'prop', 'coref', 'name', 'sense']
In [11]: s = o[0]
Loading banks for all@wsj@nw@en@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] ....... 233 trees in the treebank
reading the coreference bank [coref] .......
Enriching parse with coref ...

reading the sense bank [sense] .......
Enriching parse with sense ...
....
reading the name bank [name]....... Enriching parse with name ...
....
reading the parallel bank [parallel] ...keys: ['parse', 'prop', 'coref', 'name', 'sense']

reading the proposition bank [prop] .......
Enriching parse with prop ...
....
reading the speaker bank [speaker] ...keys: ['parse', 'prop', 'coref', 'name', 'sense']

Not enriching parse with speaker because we have no documents

Pradhan, Xue
OntoNotes: The 90% Solution
Inside the subcorpus

In [12]: s
Out[12]: subcorpus instance, id=all@wsj@nw@en@on, banks:
[ coref] : gold@all@wsj@nw@en@on
[document] : gold@all@wsj@nw@en@on
[ name] : gold@all@wsj@nw@en@on
[parallel] : gold@all@wsj@nw@en@on
[ parse] : gold@all@wsj@nw@en@on
[ prop] : gold@all@wsj@nw@en@on
[ sense] : gold@all@wsj@nw@en@on
[ speaker] : gold@all@wsj@nw@en@on

Accessing the same again does not read from the disk because
it uses weakref

In [14]: s = o[0]
In [15]: Pradhan, Xue

OntoNotes: The 90% Solution
Inside the subcorpus

In [12]: s
Inside the subcorpus

In [12]: s
Out[12]:
subcorpus instance, id=all@wsj@nw@en@on, banks:
[  coref] : gold@all@wsj@nw@en@on
[document] : gold@all@wsj@nw@en@on
[  name] : gold@all@wsj@nw@en@on
[parallel] : gold@all@wsj@nw@en@on
[  parse] : gold@all@wsj@nw@en@on
[  prop] : gold@all@wsj@nw@en@on
[  sense] : gold@all@wsj@nw@en@on
[ speaker] : gold@all@wsj@nw@en@on
Inside the subcorpus

```python
In [12]: s
Out[12]:
subcorpus instance, id=all@wsj@nw@en@on, banks:
[  coref] : gold@all@wsj@nw@en@on
[document] : gold@all@wsj@nw@en@on
[ name] : gold@all@wsj@nw@en@on
[parallel] : gold@all@wsj@nw@en@on
[ parse] : gold@all@wsj@nw@en@on
[ prop] : gold@all@wsj@nw@en@on
[ sense] : gold@all@wsj@nw@en@on
[ speaker] : gold@all@wsj@nw@en@on

Accessing the same again does not read from the disk because it uses weakref
```
Inside the subcorpus

In [12]: s
Out[12]:
subcorpus instance, id=all@wsj@nw@en@on, banks:
[  coref] : gold@all@wsj@nw@en@on
[document] : gold@all@wsj@nw@en@on
[  name] : gold@all@wsj@nw@en@on
[parallel] : gold@all@wsj@nw@en@on
[  parse] : gold@all@wsj@nw@en@on
[  prop] : gold@all@wsj@nw@en@on
[  sense] : gold@all@wsj@nw@en@on
[ speaker] : gold@all@wsj@nw@en@on

Accessing the same again does not read from the disk because it uses *weakref*

In [14]: s = o[0]
Inside the subcorpus

In [12]: s
Out[12]:
subcorpus instance, id=all@wsj@nw@en@on, banks:
    [  coref] : gold@all@wsj@nw@en@on
    [document] : gold@all@wsj@nw@en@on
    [  name] : gold@all@wsj@nw@en@on
    [parallel] : gold@all@wsj@nw@en@on
    [  parse] : gold@all@wsj@nw@en@on
    [  prop] : gold@all@wsj@nw@en@on
    [ sense] : gold@all@wsj@nw@en@on
    [ speaker] : gold@all@wsj@nw@en@on

- Accessing the same again does not read from the disk because it uses weakref

In [14]: s = o[0]
In [15]
Exploring Coreference Data

In [13]: c = s["coref"]

In [15]: c

Out[15]: coreference bank instance, id=gold@all@wsj@nw@en@on, documents:
[0] : nw/wsj/00/wsj0020@all@wsj@nw@en@on
[1] : nw/wsj/00/wsj0037@all@wsj@nw@en@on
[2] : nw/wsj/00/wsj0049@all@wsj@nw@en@on
[3] : nw/wsj/00/wsj0089@all@wsj@nw@en@on
Exploring Coreference Data

In [13]: c_bank = s["coref"]
Exploring Coreference Data

```
In [13]: c_bank = s["coref"]

In [15]: c_bank
```
Exploring Coreference Data

In [13]: c_bank = s['coref']

In [15]: c_bank
Out[15]:
coreference_bank instance, id=gold@all@wsj@nw@en@on, documents:
[0] : nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : nw/wsj/00/wsj_0037@all@wsj@nw@en@on
[2] : nw/wsj/00/wsj_0049@all@wsj@nw@en@on
[3] : nw/wsj/00/wsj_0089@all@wsj@nw@en@on
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution

In [17]:
Out[17]:
document, id=nw/wsj/00/wsj0020@all@wsj@nw@en@on, coreference
chains:
[ 0] : APPOS@000-52@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 1] : APPOS@000-57@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 2] : IDENT@000-10@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 3] : IDENT@000-12@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 4] : IDENT@000-25@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 5] : IDENT@000-2@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 6] : IDENT@000-30@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 7] : IDENT@000-33@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 8] : IDENT@000-36@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[ 9] : IDENT@000-38@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
...
...
[22] : IDENT@000-7@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
[23] : IDENT@000-9@000@nw/wsj/00/wsj0020@all@wsj@nw@en@on
In [17]: c_doc
In [17]: c_doc
Out[17]:
coreference_document, id=nw/wsj/00/wsj_0020@all@wsj@nw@en@on, coreference_chains:

[  0] : APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  1] : APPOS@000-57@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  2] : IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  3] : IDENT@000-12@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  4] : IDENT@000-25@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  5] : IDENT@000-2@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  6] : IDENT@000-30@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  7] : IDENT@000-33@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  8] : IDENT@000-36@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[  9] : IDENT@000-38@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
... 
... 
[22] : IDENT@000-7@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[23] : IDENT@000-9@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
Pradhan, Xue

OntoNotes: The 90% Solution
In [21]: c_chain = c_doc[0]
In [21]: c_chain = c_doc[0]
In [22]: c_chain
In [21]: c_chain = c_doc[0]
In [22]: c_chain
Out[22]:
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [21]: c_chain = c_doc[0]
In [22]: c_chain
Out[22]:
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on

In [24]: c_link_0 = c_chain[0]
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [21]: c_chain = c_doc[0]
In [22]: c_chain
Out[22]:
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on

In [24]: c_link_0 = c_chain[0]
In [25]: c_link_1 = c_chain[1]
In [21]: c_chain = c_doc[0]
In [22]: c_chain
Out[22]:
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on

In [24]: c_link_0 = c_chain[0]
In [25]: c_link_1 = c_chain[1]
In [26]: c_link_0
In [21]: c_chain = c_doc[0]

In [22]: c_chain

Out[22]:
coreference chain instance, id=APPOS@000-52@0000nw/ws0j/00/ws0j_0020@all@wsj@nw@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@0000nw/ws0j/00/ws0j_0020@all@wsj@nw@en@on
[1] : HEAD@1:6:14@APPOS@000-52@0000nw/ws0j/00/ws0j_0020@all@wsj@nw@en@on

In [24]: c_link_0 = c_chain[0]

In [25]: c_link_1 = c_chain[1]

In [26]: c_link_0

Out[26]: <coreference_link object: id: ATTRIB@1:2:4@APPOS@000-52@0000nw/ws0j/00/ws0j_0020@all@wsj@...; type: ATTRIB --- ‘five other countries’>
In [21]: c_chain = c_doc[0]
In [22]: c_chain
Out[22]:
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on

In [24]: c_link_0 = c_chain[0]
In [25]: c_link_1 = c_chain[1]

In [26]: c_link_0
Out[26]: <coreference_link object: id: ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@en@on; type: ATTRIB --- ‘five other countries’>

In [27]: c_link_1
In [21]: c_chain = c_doc[0]
In [22]: c_chain
Out[22]:
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on, links:
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on

In [24]: c_link_0 = c_chain[0]

In [25]: c_link_1 = c_chain[1]

In [26]: c_link_0
Out[26]: <coreference_link object: id: ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@...; type: ATTRIB --- ‘five other countries’>

In [27]: c_link_1
Out[27]: <coreference_link object: id: HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/wsj_0020@all@wsj@...; type: HEAD --- ‘China , Thailand , India , Brazil and Mexico’>
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [32]: c
Out[32]: coreference chain instance, id=IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on, links:
[ 0] : IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 1] : IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 2] : IDENT@3:10:10@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 3] : IDENT@4:0:0@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 4] : IDENT@6:0:1@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 5] : IDENT@7:19:19@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 6] : IDENT@10:35:36@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 7] : IDENT@17:0:1@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 8] : IDENT@18:0:0@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[ 9] : IDENT@19:5:6@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on
[10] : IDENT@20:3:4@IDENT@000-10@000@nw/wsj/00/wsj
0020@all@wsj@nw@en@on

In [38]: c
link0 = c
chain[0]
In [39]: c
link1 = c
chain[1]
In [40]: c
link2 = c
chain[2]
In [32]: c_chain
In [32]: c_chain
Out[32]:
coreference chain instance, id=IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on, links:
[ 0] : IDENT@1:34:38@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 1] : IDENT@3:0:1@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 2] : IDENT@3:10:10@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 3] : IDENT@4:0:0@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 4] : IDENT@6:0:1@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 5] : IDENT@7:19:19@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 6] : IDENT@10:35:36@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 7] : IDENT@17:0:1@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 8] : IDENT@18:0:0@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[ 9] : IDENT@19:5:6@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on
[10] : IDENT@20:3:4@IDENT@000-10@000@nw/ws/00/ws/0020@all/ws@nw@en@on

Pradhan, Xue
OntoNotes: The 90% Solution
In [32]: c_chain
Out[32]:
coreference chain instance, id=IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on, links:
[ 0] : IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 1] : IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 2] : IDENT@3:10:10@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 3] : IDENT@4:0:0@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 4] : IDENT@6:0:1@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 5] : IDENT@7:19:19@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 6] : IDENT@10:35:36@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 7] : IDENT@17:0:1@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 8] : IDENT@18:0:0@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[ 9] : IDENT@19:5:6@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on
[10] : IDENT@20:3:4@IDENT@000-10@000@nw/wsj/00/wsja_0020@all@wsj@nw@en@on

In [38]: c_link_0 = c_chain[0]
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [32]: c_chain
Out[32]:
coreference chain instance, id=IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on, links:
  [ 0] : IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 1] : IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 2] : IDENT@3:10:10@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 3] : IDENT@4:0:0@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 4] : IDENT@6:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 5] : IDENT@7:19:19@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 6] : IDENT@10:35:36@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 7] : IDENT@17:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 8] : IDENT@18:0:0@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
  [ 9] : IDENT@19:5:6@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[10] : IDENT@20:3:4@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on

In [38]: c_link_0 = c_chain[0]

In [39]: c_link_1 = c_chain[1]
In [32]: c_chain

Out[32]:

coreference chain instance, id=IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on, links:
[0] : IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[2] : IDENT@3:10:10@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[3] : IDENT@4:0:0@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[4] : IDENT@6:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[5] : IDENT@7:19:19@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[6] : IDENT@10:35:36@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[7] : IDENT@17:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[8] : IDENT@18:0:0@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[9] : IDENT@19:5:6@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[10]: IDENT@20:3:4@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@nw@en@on

In [38]: c_link_0 = c_chain[0]

In [39]: c_link_1 = c_chain[1]

In [40]: c_link_2 = c_chain[2]
In [41]: c_link_0
Pradhan, Xue

OntoNotes: The 90% Solution
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [41]: c_link_0
Out[41]: <coreference_link object: id: IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...; type: IDENT --- ‘U.S. Trade Representative Carla Hills’>

In [42]: c_link_1
In [41]: c_link_0
Out[41]: <coreference_link object: id: IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...; type: IDENT --- ‘U.S. Trade Representative Carla Hills’>

In [42]: c_link_1
Out[42]: <coreference_link object: id: IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...; type: IDENT --- ‘Mrs. Hills’>
In [41]: c_link_0
Out[41]: <coreference_link object: id: IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...; type: IDENT --- ‘U.S. Trade Representative Carla Hills’>

In [42]: c_link_1
Out[42]: <coreference_link object: id: IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...; type: IDENT --- ‘Mrs. Hills’>

In [43]: c_link_2
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [41]: c_link_0
Out[41]: <coreference_link object: id: IDENT@1:34:38@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...;
type: IDENT --- ‘U.S. Trade Representative Carla Hills’>

In [42]: c_link_1
Out[42]: <coreference_link object: id: IDENT@3:0:1@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...;
type: IDENT --- ‘Mrs. Hills’>

In [43]: c_link_2
Out[43]: <coreference_link object: id: IDENT@3:10:10@IDENT@000-10@000@nw/wsj/00/wsj_0020@all@wsj@...;
type: IDENT --- ‘she’>
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [45]: c_link_0.tab
In [45]: `c_link_0.[tab]`

c_link_0.__class_ff_
c_link_0.__delattr__
c_link_0.__dict__
c_link_0.__doc__
c_link_0.__getattribute__
c_link_0.__hash__
c_link_0.__init__
c_link_0.__module__
c_link_0.__new__
c_link_0.__reduce__
c_link_0.__reduce_ex__
c_link_0.__repr__
c_link_0.__setattr__
c_link_0.__str__
c_link_0.__weakref__
c_link_0.end_leaf
c_link_0.end_token_index
c_link_0.end_word_index
c_link_0.get_end_leaf
c_link_0.get_end_token_index
c_link_0.get_end_word_index
c_link_0.get_string
c_link_0.get_subtree_id
c_link_0.get_sentence_index
c_link_0.set_end_leaf
c_link_0.set_end_token_index
c_link_0.set_end_word_index
c_link_0.set_start_leaf
c_link_0.set_start_token_index
c_link_0.set_start_word_index
c_link_0.string

Pradhan, Xue

OntoNotes: The 90% Solution
In [45]: c_link_0.[tab]

c_link_0.__class_ff_
c_link_0.__delattr__
c_link_0.__dict__
c_link_0.__doc__
c_link_0.__getattribute__
c_link_0.__hash__
c_link_0.__init__
c_link_0.__module__
c_link_0.__new__
c_link_0.__reduce__
c_link_0.__reduce_ex__
c_link_0.__repr__
c_link_0.__setattr__
c_link_0.__str__
c_link_0.__weakref__
c_link_0.__end_leaf
c_link_0.__end_token_index
c_link_0.__end_word_index
c_link_0._enrich_tree
c_link_0.id
c_link_0.overlaps
c_link_0.primary_end_index
c_link_0.primary_start_index
c_link_0.secondary_start_index
c_link_0.secondary_token_index
c_link_0.sql_create_statement
c_link_0.sql_insert_statement
c_link_0.valid
c_link_0.write_to_db
c.link_0._get_subtree_id
c.link_0._sentence_index
c.link_0._set_end_leaf
c.link_0._set_start_leaf
c.link_0._set_string
c.link_0._start_leaf
c.link_0._start_token_index
c.link_0._start_word_index
c.link_0._string
c.link_0._subtree_id
c.link_0.sql_table_name
c.link_0.start_leaf
c.link_0.start_token_index
c.link_0.start_word_index
c.link_0.string
c.link_0.subtree
c.link_0.subtree_id
c.link_0.type
c.link_0.copy_to_different_trees
c.link_0.coreference_chain
c.link_0.coreference_chain_id
c.link_0.coreference_chain_id
c.link_0.end_leaf

Pradhan, Xue | OntoNotes: The 90% Solution
<table>
<thead>
<tr>
<th>Challenges with Multiple Layers of Annotation</th>
<th>Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>Creating ontonotes</td>
</tr>
<tr>
<td>Raw Data</td>
<td>Exploring Various Layers</td>
</tr>
<tr>
<td>Database Design</td>
<td>Exploring Parallel Connections</td>
</tr>
<tr>
<td>Python API Design</td>
<td>Advanced Topics</td>
</tr>
<tr>
<td>Data Access</td>
<td>Cross-Layer Query</td>
</tr>
</tbody>
</table>

In [45]: c

Out[45]:

(NP-SBJ (NML (NNP U.S.)
(NNP Trade)
(NNP Representative))

(NNP Carla)

(NNP Hills))

In [46]: c

Out[46]:

34:2@1@nw/wsj/00/wsj0020@all@wsj@nw@en@on

In [47]: c

Out[47]:

IDENT

Pradhan, Xue

OntoNotes: The 90% Solution
In [45]: c_link_0.subtree
In [45]: c_link_0.subtree
Out[45]:

(NP-SBJ (NML (NNP U.S.)
    (NNP Trade)
    (NNP Representative))
    (NNP Carla)
    (NNP Hills))
In [45]: c_link_0.subtree

Out[45]:

(NP-SBJ (NML (NNP U.S.)
    (NNP Trade)
    (NNP Representative)))
    (NNP Carla)
    (NNP Hills))

In [46]: c_link_0.subtree_id
In [45]: c_link_0.subtree
Out[45]:
(NP-SBJ (NML (NNP U.S.))
  (NNP Trade)
  (NNP Representative))
  (NNP Carla)
  (NNP Hills))

In [46]: c_link_0.subtree_id
Out[46]: '34:2@1@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [45]: c_link_0.subtree
Out[45]:
(NP-SBJ (NML (NNP U.S.)
  (NNP Trade)
  (NNP Representative))
  (NNP Carla)
  (NNP Hills))

In [46]: c_link_0.subtree_id
Out[46]: '34:2@1@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [47]: c_link_0.type
Pradhan, Xue

In [45]: c_link_0.subtree
Out[45]:

(NP-SBJ (NML (NNP U.S.)
  (NNP Trade)
  (NNP Representative))
  (NNP Carla)
  (NNP Hills))

In [46]: c_link_0.subtree_id
Out[46]: ‘34:2@1@nw/wsj/00/wsj_0020@all@wsj@nw@en@on’

In [47]: c_link_0.type
Out[47]: ‘IDENT’
Exploring Treebank Data
Exploring Treebank Data

In [48]: t_bank = s["parse"]
Exploring Treebank Data

In [48]: t_bank = s["parse"]

In [49]: t_bank
Exploring Treebank Data

In [48]: t_bank = s["parse"]

In [49]: t_bank

Out[49]:
treebank instance, id=gold@all@wsj@nw@en@on, documents:
[0] : nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : nw/wsj/00/wsj_0037@all@wsj@nw@en@on
[2] : nw/wsj/00/wsj_0049@all@wsj@nw@en@on
[3] : nw/wsj/00/wsj_0089@all@wsj@nw@en@on
Exploring Treebank Data

```
In [48]: t_bank = s["parse"]

In [49]: t_bank
Out[49]:
treebank instance, id=gold@all@wsj@nw@en@on, documents:
   [0] : nw/wsj/00/wsj_0020@all@wsj@nw@en@on
   [1] : nw/wsj/00/wsj_0037@all@wsj@nw@en@on
   [2] : nw/wsj/00/wsj_0049@all@wsj@nw@en@on
   [3] : nw/wsj/00/wsj_0089@all@wsj@nw@en@on

In [50]: t_doc = t_bank[0]
```
Exploring Treebank Data

In[48]: t_bank = s["parse"]

In[49]: t_bank
Out[49]:
treebank instance, id=gold@all@wsj@nw@en@on, documents:
[0]: nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1]: nw/wsj/00/wsj_0037@all@wsj@nw@en@on
[2]: nw/wsj/00/wsj_0049@all@wsj@nw@en@on
[3]: nw/wsj/00/wsj_0089@all@wsj@nw@en@on

In[50]: t_doc = t_bank[0]

In[51]: t_doc
Exploring Treebank Data

In [48]: t_bank = s['parse']

In [49]: t_bank
Out[49]:
treebank instance, id=gold@all@wsj@nw@en@on, documents:
    [0] : nw/wsj/00/wsj_0020@all@wsj@nw@en@on
    [1] : nw/wsj/00/wsj_0037@all@wsj@nw@en@on
    [2] : nw/wsj/00/wsj_0049@all@wsj@nw@en@on
    [3] : nw/wsj/00/wsj_0089@all@wsj@nw@en@on

In [50]: t_doc = t_bank[0]

In [51]: t_doc
Out[51]:
tree_document instance, id=nw/wsj/00/wsj_0020@all@wsj@nw@en@on, trees:
    [ 0] : 0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
    [ 1] : 1@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
    [ 2] : 2@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
    [ 3] : 3@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
Pradhan, Xue

OntoNotes: The 90% Solution
In [52]: t_0 = t_doc[0]
In [52]: t_0 = t_doc[0]

In [56]: t_0

Pradhan, Xue

OntoNotes: The 90% Solution
In [52]: t_0 = t_doc[0]

In [56]: t_0

Out[56]:
<on.corpora.tree object id=0nw/wsj/00/wsj_0020@all@wsj@nw@en@on value=<
(TOP (S (NP-SBJ-1 (DT The)
  (NNP U.S.))
  (, ,)
  (S-ADV (NP-SBJ (-NONE- *PRO*-1))
    (VP (VBG claiming)
      (NP (NP (DT some)
        (NN success))
      (PP-LOC (IN in)
        (NP (PRP$ its)
          (NN trade)
          (NN diplomacy))))))
  (, ,)
  (VP (VBD removed)
    (NP (NP (NNP South)
      (NNP Korea))
      (, ,)
      (NP (NNP Taiwan))
    (CC and)
      (NP (NNP Saudi)
        (NNP Arabia)))
    (PP-CLR (IN from)
      (NP (NP (DT a)
        (NP (NNP Pradhan, Xue
          (NNP Xue
          (NNP OntoNotes: The 90% Solution)
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue

OntoNotes: The 90% Solution
In [65]: for leaf in t_0.leaves():
    print leaf
    
Pradhan, Xue
OntoNotes: The 90% Solution
In [65]: for leaf in t_0.leaves():
    .....: print leaf
    .....:
    .....:
    (DT The)
(NNP U.S.)
(, ,)
(-NONE- *PRO*-1)
(VBG claiming)
(DT some)
(NN success)
(IN in)
(PRPS its)
(NN trade)
(NN diplomacy)
...
...
...
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [19]: t_0[3:11]
In [19]: t_0[3:11]
Out[19]:
(S-ADV (NP-SBJ (-NONE- *PRO*-1))
   (VP (VBG claiming)
      (NP (NP (DT some)
        (NN success))
      (PP-LOC (IN in)
        (NP (PRP$ its)
          (NN trade)
          (NN diplomacy)))))

Pradhan, Xue
OntoNotes: The 90% Solution
In [19]: t_0[3:11]
Out[19]:
(S-ADV (NP-SBJ (-NONE- *PRO*-1))
  (VP (VBG claiming)
    (NP (NP (DT some)
        (NN success))
     (PP-LOC (IN in)
      (NP (PRP$ its)
       (NN trade)
       (NN diplomacy))))))

In [20]: t_0[3:10]
Challenges with Multiple Layers of Annotation

Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [19]: t_0[3:11]
Out[19]:
(S-ADV (NP-SBJ (-NONE- *PRO*-1))
  (VP (VBG claiming)
    (NP (NP (DT some)
      (NN success))
    (PP-LOC (IN in)
      (NP (PRP$ its)
        (NN trade)
        (NN diplomacy))))))

In [20]: t_0[3:10]
Out[20]:
[(-NONE- *PRO*-1),
  (VBG claiming),
  (DT some),
  (NN success),
  (IN in),
  (PRP$ its),
  (NN trade)]

Pradhan, Xue
OntoNotes: The 90% Solution
Exploring Proposition Data
Exploring Proposition Data

In [87]: p = []
In [87]: p = []

In [88]: for leaf in t_0.leaves():
   ...:     if(leaf.proposition != None):
   ...:         p.append(leaf.proposition)
   
   ....:
   ....:
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [95]: p[3]
In [95]: p[3]
Out[95]:

proposition:

id : 29@0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
doc_id : nw/wsj/00/wsj_0020@all@wsj@nw@en@on
tree_id : 0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on
frame : watch.01
enc_prop : wsj_0020@... 0 29 ... watch.01 ----- 29:0-rel 26:1-ARG0 30:0*25:1-ARG1 ... 24:1*25:1-LINK-SLC
predicate:
< predicate_analogue : id: watch.01@v@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘29:0-rel’
  < predicate : id: 0@watch.01@v@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘29:0’
  < predicate_node : id: 0@0@watch.01@v@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘29:0’>>>
arguments:
< argument_analogue : id: 0@ARG0@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘26:1-ARG0’
  < argument : id: 0@0@ARG0@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘26:1’
  < argument_node : id: 0@0@0@ARG0@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘26:1’>>>
...
...
lights:
< link_analogue : id: 0@LINK-SLC@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘24:1*25:1-LINK-SLC’
  < link : id: 0@0@LINK-SLC@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘24:1’
  < link_node : id: 0@0@0@0@LINK-SLC@29@0@nw/wsj/00/wsj_0020@...; enc_self: ‘24:1’>>, Pradhan, Xue
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [98]: predicate = p[3].predicate

In [99]: predicate.

class predicate.add predicate.get

primary

predicate.
delattr

predicate.analogue

type predicate.id

dictionary predicate.children predicate.index

in

parent

dictionary predicate.
doc

to
different
trees

predicate.lemma

getattribute

predicate.document

id predicate.parent

getitem

predicate.enc

self

predicate.pb

sense

num

gethash

predicate.enc

self

type

primary

predicate

init

predicate.enrich
tree

predicate.
len

predicate.
get

index

doctrine

index

of

predicate.
sentence

index

In [99]: predicate.lemma

Out[99]: u'watch'

In [100]: predicate.tree

Out[100]: u'0@nw/wsj/00/wsj 0020@all@wsj@nw@en@on'

In [101]: predicate.token

Out[101]: 29

Pradhan, Xue

OntoNotes: The 90% Solution
In [98]: predicate = p[3].predicate
In [98]: predicate = p[3].predicate

In [99]: predicate.

[tab]

In [99]: predicate.tab
In [98]: predicate = p[3].predicate

In [99]: predicate.

predicate.__class__
predicate.__delattr__
predicate.__dict__
predicate.__getattr__
predicate.__getattribute__
predicate.__getitem__
predicate.__hash__
predicate.__init__
predicate.__len__
predicate.add
predicate.analogue_type
predicate.children
predicate.copy_to_different_trees
predicate.document_id
predicate.enc_self
predicate.enc_self_type
predicate.enrich_tree
predicate.get_index_of
predicate.get_primary_predicate
predicate.id
predicate.index_in_parent
predicate.lemma
predicate.parent
predicate.pb_sense_num
predicate.primary_predicate
predicate.proposition
predicate.sentence_index

In [99]: predicate.lemma
Out[99]: u'watch'

In [100]: predicate.tree
Out[100]: '0@nw/wsj/00/wsj0020@all@wsj@nw@en@on'

In [101]: predicate.token
Out[101]: 29
In [98]: predicate = p[3].predicate

In [99]: predicate.

```
predicate.__class__
predicate.__delattr__
predicate.__dict__
predicate.__doc__
predicate.__getattribute__
predicate.__getitem__
predicate.__hash__
predicate.__init__
predicate.__len__
```

```
predicate.add
predicate.analogue_type
predicate.children
predicate.copy_to_different_trees
predicate.document_id
predicate.enc_self
predicate.enc_self_type
predicate.enrich_tree
predicate.get_index_of
```

```
predicate.get_primary_predicate
predicate.id
predicate.index_in_parent
predicate.lemma
predicate.parent
predicate.pb_sense_num
predicate.primary_predicate
predicate.proposition
predicate.sentence_index
```

In [99]: predicate.lemma

```
Out[99]: u'watch'
```

In [100]: predicate.tree
id

```
Out[100]: '0@nw/wsj/00/wsj
0020@all@wsj@nw@en@on'
```

In [101]: predicate.token

```
Out[101]: 29
```
Challenges with Multiple Layers of Annotation

Architecture
Raw Data
Database Design
Python API Design
Data Access

Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [98]: predicate = p[3].predicate

In [99]: predicate.

predicate.__class__
predicate.__delattr__
predicate.__dict__
predicate.__delitem__
predicate.__getattribute__
predicate.__getitem__
predicate.__hash__
predicate.__init__
predicate.__len__
predicate.add
predicate.analogue_type
predicate.children
predicate.copy_to_different_trees
predicate.document_id
predicate.enc_self
predicate.enc_self_type
predicate.enrich_tree
predicate.get_attribute
predicate.get_index_of
predicate.get_primary_predicate
predicate.id
predicate.index_in_parent
predicate.lemma
predicate.parent
predicate.pb_sense_num
predicate.primary_predicate
predicate.proposition
predicate.sentense_index

In [99]: predicate.lemma

Out[99]: u'watch'
In [98]: predicate = p[3].predicate

In [99]: predicate.

```
predicate.__class__
predicate.__delattr__
predicate.__dict__
predicate.__getattribute__
predicate.__getitem__
predicate.__hash__
predicate.__init__
predicate.__len__
predicate.add
predicate.analogue_type
predicate.children
predicate.copy_to_different_trees
predicate.document_id
predicate.enc_self
predicate.enc_self_type
predicate.enrich_tree
predicate.get_index_of
predicate.get_primary_predicate
predicate.id
predicate.index_in_parent
predicate.lemma
predicate.parent
predicate.pb_sense_num
predicate.primary_predicate
predicate.proposition
predicate.set
predicate.tree_id
```

In [99]: predicate.lemma
Out[99]: u'watch'

In [100]: predicate.tree_id
In [98]: predicate = p[3].predicate

In [99]: predicate.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>class</strong></td>
<td>Class of predicate</td>
</tr>
<tr>
<td><strong>delattr</strong></td>
<td>Delete attribute</td>
</tr>
<tr>
<td><strong>dict</strong></td>
<td>Dictionary of predicate attributes</td>
</tr>
<tr>
<td><strong>doc</strong></td>
<td>Documentation of predicate</td>
</tr>
<tr>
<td><strong>getattribute</strong></td>
<td>Get attribute</td>
</tr>
<tr>
<td><strong>getitem</strong></td>
<td>Get item by key</td>
</tr>
<tr>
<td><strong>hash</strong></td>
<td>Hash of predicate</td>
</tr>
<tr>
<td><strong>init</strong></td>
<td>Initialize predicate</td>
</tr>
<tr>
<td><strong>len</strong></td>
<td>Length of predicate</td>
</tr>
<tr>
<td>add</td>
<td>Add predicate</td>
</tr>
<tr>
<td>analogue_type</td>
<td>analogue type of predicate</td>
</tr>
<tr>
<td>children</td>
<td>Children of predicate</td>
</tr>
<tr>
<td>copy_to_different_trees</td>
<td>Copy to different trees</td>
</tr>
<tr>
<td>document_id</td>
<td>Document id of predicate</td>
</tr>
<tr>
<td>enc_self</td>
<td>Encoded self of predicate</td>
</tr>
<tr>
<td>enc_self_type</td>
<td>Encoded self type of predicate</td>
</tr>
<tr>
<td>enrich_tree</td>
<td>Enrich tree</td>
</tr>
<tr>
<td>get_index_of</td>
<td>Get index of</td>
</tr>
<tr>
<td>get_primary_predicate</td>
<td>Get primary predicate</td>
</tr>
<tr>
<td>id</td>
<td>Id of predicate</td>
</tr>
<tr>
<td>index_in_parent</td>
<td>Index in parent</td>
</tr>
<tr>
<td>lemma</td>
<td>Lemma of predicate</td>
</tr>
<tr>
<td>parent</td>
<td>Parent of predicate</td>
</tr>
<tr>
<td>pb_sense_num</td>
<td>Primary sense number</td>
</tr>
<tr>
<td>primary_predicate</td>
<td>Primary predicate</td>
</tr>
<tr>
<td>proposition</td>
<td>Proposition of predicate</td>
</tr>
<tr>
<td>sentence_index</td>
<td>Sentence index of predicate</td>
</tr>
</tbody>
</table>

In [99]: predicate.lemma
Out[99]: u'watch'

In [100]: predicate.tree_id
Out[100]: '0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'
In [98]: predicate = p[3].predicate

In [99]: predicate.

```python
predicate.__class__
predicate.__delattr__
predicate.__dict__
predicate.__getattribute__
predicate.__getitem__
predicate.__hash__
predicate.__init__
predicate.__len__
predicate.add
predicate.analogue_type
predicate.children
predicate.copy_to_different_trees
predicate.document_id
predicate.enc_self
predicate.enc_self_type
predicate.enrich_tree
predicate.get_index_of
predicate.get_primary_predicate
predicate.id
predicate.index_in_parent
predicate.lemma
predicate.parent
predicate.pb_sense_num
predicate.primary_predicate
predicate.proposition
predicate.sentence_index
```

In [99]: predicate.lemma
Out[99]: u'watch'

In [100]: predicate.tree_id
Out[100]: '0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [101]: predicate.token_index
In [98]: predicate = p[3].predicate

In [99]: predicate.

```python
predicate.__class__
predicate.add
predicate.analogue_type
predicate.children
predicate.copy_to_different_trees
predicate.document_id
predicate.encoded
predicate.encoded_type
predicate.enrich_tree
predicate.get_index_of
predicate.get_attribute
predicate.getitem
predicate.hash
predicate.lemma
predicate.len
predicate.encoded
predicate.get
predicate.index
predicate.getitem
predicate.encoded
predicate.getitem
predicate.index
predicate.getitem
predicate.encoded
predicate.getitem
predicate.index
predicate.getitem
```

In [99]: predicate.lemma
Out[99]: u'watch'

In [100]: predicate.tree_id
Out[100]: '0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [101]: predicate.token_index
Out[101]: 29
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query
In [102]: predicate.document_id
In [102]: predicate.document_id
Out[102]: ‘nw/.wsj/00/wsj_0020@all@wsj@nw@en@on’
In [102]: predicate.document_id
Out[102]: 'nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [103]: predicate.sentence_index

Pradhan, Xue
OntoNotes: The 90% Solution
In [102]: `predicate.document_id`
Out[102]: 'nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [103]: `predicate.sentence_index`
Out[103]: 0
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [102]: predicate.document_id
Out[102]: 'nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [103]: predicate.sentence_index
Out[103]: 0

In [104]: predicate.type
In [102]: predicate.document_id
Out[102]: ‘nw/wsj/00/wsj_0020@all@wsj@nw@en@on’

In [103]: predicate.sentence_index
Out[103]: 0

In [104]: predicate.type
Out[104]: u‘v’
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue

OntoNotes: The 90% Solution
In [113]: link_analogue = proposition.link_analogues
In [113]: link_analogue = proposition.link.analogues

In [120]: link_analogue[0][0][0]
In [113]: link_analogue = proposition.link_analogues

In [120]: link_analogue[0][0][0]
Out[120]: <link_node id: 00000LINK-SLC029@0..wsj_0020@...@on; enc_self: '24:1'>
In [113]: link_analogue = proposition.link_analogues

In [120]: link_analogue[0][0][0]
Out[120]: <link_node id: 0@0@0@LINK-SLC@29@0@..wsj_0020@...@on; enc_self: ‘24:1’>

In [121]: link_node = link_analogue[0][0][0]
In [113]: `link_analogue = proposition.link_analogues`

In [120]: `link_analogue[0][0][0]`

Out[120]: `<link_node id: 00000LINK-SLC02900..wsj_00200...@on; enc_self: ‘24:1’>`

In [121]: `link_node = link_analogue[0][0][0]`

In [122]: `link_node.type`
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [113]: link_analogue = proposition.link_analogues

In [120]: link_analogue[0][0][0]
Out[120]: <link node id: 0@0@0@LINK-SLC@29@0@..wsj_0020@...@on; enc_self: ‘24:1’>

In [121]: link_node = link_analogue[0][0][0]

In [122]: link_node.type
Out[122]: u‘LINK-SLC’
In [113]: `link_analogue = proposition.link_analogues`

In [120]: `link_analogue[0][0][0]`
Out[120]: `<link node id: 00000LINK-SLC29000wsj_002000on; enc_self: ‘24:1’>`

In [121]: `link_node = link_analogue[0][0][0]`

In [122]: `link_node.type`
Out[122]: `u‘LINK-SLC’`

In [123]: `link_node.subtree`
In [113]: link_analogue = proposition.link_analogues

In [120]: link_analogue[0][0][0]
Out[120]: <link_node id: 0@0@0@LINK-SLC@29@0@...@on; enc_self: ‘24:1’>

In [121]: link_node = link_analogue[0][0][0]

In [122]: link_node.type
Out[122]: u‘LINK-SLC’

In [123]: link_node.subtree
Out[123]: (NP (NNS countries))
In [125]: link.node.subtree.start
Pradhan, Xue

OntoNotes: The 90% Solution
In [125]: link.node.subtree.start
Out[125]: 24

In [126]: link.node.subtree.end
In [125]: link.node.subtree.start
Out[125]: 24

In [126]: link.node.subtree.end
Out[126]: 25
In [125]: link_node.subtree.start
Out[125]: 24

In [126]: link_node.subtree.end
Out[126]: 25

In [128]: link_node.id
In [125]: link_node.subtree.start
Out[125]: 24

In [126]: link_node.subtree.end
Out[126]: 25

In [128]: link_node.id
Out[128]: u'0@0@0@LINK-SLC@29@0@nw/wsj/00/wnj_0020@all@wnj@nw@en@on'
In [125]: `link_node.subtree.start`
Out[125]: 24

In [126]: `link_node.subtree.end`
Out[126]: 25

In [128]: `link_node.id`
Out[128]: `u'00000@LINK-SLC@2900@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'`

In [134]: `link_node.subtree.get_word[tab]`
In [125]: `link_node.subtree.start`
Out[125]: 24

In [126]: `link_node.subtree.end`
Out[126]: 25

In [128]: `link_node.id`
Out[128]: u'0@@0@LINK-SLC@2900@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [134]: `link_node.subtree.get_word[tab]`
`link_node.subtree.get_word_index` `link_node.subtree.get_word_string`
In [125]: link_node.subtree.start
Out[125]: 24

In [126]: link_node.subtree.end
Out[126]: 25

In [128]: link_node.id
Out[128]: u'0@0@0@LINK-SLC@29@0@nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [134]: link_node.subtree.get_word[tab]
link_node.subtree.get_word_index link_node.subtree.get_word_string

In [134]: link_node.subtree.get_word_string()
In [125]: link_node.subtree.start
Out[125]: 24

In [126]: link_node.subtree.end
Out[126]: 25

In [128]: link_node.id
Out[128]: u'00000LINK-SLC02900nw/wsj/00/wsj_0020@all@wsj@nw@en@on'

In [134]: link_node.subtree.get_word(tab)
link_node.subtree.get_word_index   link_node.subtree.get_word_string

In [134]: link_node.subtree.get_word_string()
Out[134]: u'countries'
Exploring Senses

In [177]:
   s

In [178]:
   s

Out[178]:
   sense
   bank instance, id=gold@all@wsj@nw@en@on, documents:
   [0] : nw/wsj/00/wsj
   0020@all@wsj@nw@en@on
   [1] : nw/wsj/00/wsj
   0037@all@wsj@nw@en@on
   [2] : nw/wsj/00/wsj
   0049@all@wsj@nw@en@on
   [3] : nw/wsj/00/wsj
   0089@all@wsj@nw@en@on

In [179]:
   s
doc
   0 = s
   bank[0]

In [180]:
   s
doc
   0

Out[180]:
   senses
tagged
document instance, id=nw/wsj/00/wsj
   0020@all@wsj@nw@en@on, on
   senses:
   [ 0] : claim.2@v@3@0@nw/wsj/00/wsj
   0020@all@wsj@nw@en@on
   [ 1] : success.2@n@5@0@nw/wsj/00/wsj
   0020@all@wsj@nw@en@on
   [ 2] : trade.1@n@8@0@nw/wsj/00/wsj
   0020@all@wsj@nw@en@on
   [ 3] : remove.1@v@11@0@nw/wsj/00/wsj
   0020@all@wsj@nw@en@on
   [ 4] : list.1@n@21@0@nw/wsj/00/wsj

Pradhan, Xue

OntoNotes: The 90% Solution
Exploring Senses

In [177]: s_bank = s["sense"]
Exploring Senses

In [177]: s_bank = s["sense"]
In [178]: s_bank
Exploring Senses

In [177]: s_bank = s["sense"]
In [178]: s_bank
Out[178]:

sense_bank instance, id=gold@all@wsj@nw@en@on, documents:
[0] : nw/wsj/00/wnj_0020@all@wsj@nw@en@on
[1] : nw/wnj/00/wnj_0037@all@wsj@nw@en@on
[2] : nw/wnj/00/wnj_0049@all@wsj@nw@en@on
[3] : nw/wnj/00/wnj_0089@all@wsj@nw@en@on
Exploring Senses

In [177]: s_bank = s["sense"]
In [178]: s_bank
Out[178]:

sense_bank instance, id=gold@all@wsj@nw@en@on, documents:
[0] : nw/wsj/00/wsj_0020@all@wsj@nw@en@on
[1] : nw/wsj/00/wsj_0037@all@wsj@nw@en@on
[2] : nw/wsj/00/wsj_0049@all@wsj@nw@en@on
[3] : nw/wsj/00/wsj_0089@all@wsj@nw@en@on

In [179]: s_doc_0 = s_bank[0]
Exploring Senses

In [177]: s_bank = s["sense"]
In [178]: s_bank
Out[178]:

sense_bank instance, id=gold@all@wsj@nw@en@on, documents:
[0] : nw/ws/j/00/wsj_0020@all@wsj@nw@en@on
[1] : nw/ws/j/00/wsj_0037@all@wsj@nw@en@on
[2] : nw/ws/j/00/wsj_0049@all@wsj@nw@en@on
[3] : nw/ws/j/00/wsj_0089@all@wsj@nw@en@on

In [179]: s_doc_0 = s_bank[0]

In [180]: s_doc_0
Exploring Senses

In [177]: s_bank = s["sense"]

In [178]: s_bank

Out[178]:
sense_bank instance, id=gold@all@wsj@nw@en@on, documents:
    [0] : nw/hsj/00/hsj_0020@all@wsj@nw@en@on
    [1] : nw/hsj/00/hsj_0037@all@wsj@nw@en@on
    [2] : nw/hsj/00/hsj_0049@all@wsj@nw@en@on
    [3] : nw/hsj/00/hsj_0089@all@wsj@nw@en@on

In [179]: s_doc_0 = s_bank[0]

In [180]: s_doc_0

Out[180]:
senses_tagged_document instance, id=nw/hsj/00/hsj_0020@all@wsj@nw@en@on, on_senses:
    [ 0] : claim.2@v@3@0@nw/hsj/00/hsj_0020@all@wsj@nw@en@on
    [ 1] : success.2@n@5@0@nw/hsj/00/hsj_0020@all@wsj@nw@en@on
    [ 2] : trade.1@n@8@0@nw/hsj/00/hsj_0020@all@wsj@nw@en@on
    [ 3] : remove.1@v@11@0@nw/hsj/00/hsj_0020@all@wsj@nw@en@on
    [ 4] : list.1@n@21@0@nw/hsj/00/hsj_s0020@all@wsj@nw@en@on
Exploring Parallel Connections
Pradhan, Xue

OntoNotes: The 90% Solution
Exploring Parallel Connections

In [136]: s["parallel"]
Out[136]:
parallel_bank instance, id=gold@all@wsj@nw@en@on, documents:
  (empty)
Pradhan, Xue

OntoNotes: The 90% Solution
In [137]: c = on.common.util.load_config("config.parallel")
In [137]: c = on.common.util.load_config("config.parallel")

In [138]: o = on.ontonotes(c)
In [137]: c = on.common.util.load_config("config.parallel")

In [138]: o = on.ontonotes(c)
Loading chinese bc msnbc
........
found 1 file in the subcorpus all@msnbc@bc@ch@on
Loading english bc msnbc
........
found 1 file in the subcorpus all@msnbc@bc@en@on
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [139]: o
Out[139]:

Pradhan, Xue

OntoNotes: The 90% Solution
| In [139]: |  o |

Pradhan, Xue

OntoNotes: The 90% Solution
In [139]:
Out[139]:
ontonotes instance, id=on, subcorpora:
[0] : all@msnbc@bc@ch@on
[1] : all@msnbc@bc@en@on
Pradhan, Xue

OntoNotes: The 90% Solution
Challenges with Multiple Layers of Annotation

Exploring Parallel Connections

Pradhan, Xue
In [140]: s_0 = o[0]
Loading banks for all@msnbc@bc@ch@on: parse, coref, sense, name, parallel, prop, speaker .... reading the treebank [parse] .... 665 trees in the treebank
reading the coreference bank [coref] .... Enriching parse with coref ...
reading the sense bank [sense] .... Enriching parse with sense ...
reading the name bank [name] .... Enriching parse with name ...
reading the parallel bank [parallel] ....
In [140]: s_0 = o[0]
Loading banks for all@msnbc@bc@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] .... 665 trees in the treebank
reading the coreference bank [coref] .... Enriching parse with coref ...
reading the sense bank [sense] .... Enriching parse with sense ...
reading the name bank [name].... Enriching parse with name ...
reading the parallel bank [parallel] ....

finding original trees to prepare for parallel bank enrichment....
In [140]: s_0 = o[0]
Loading banks for all@msnbc@bc@ch@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] .... 665 trees in the treebank
reading the coreference bank [coref] .... Enriching parse with coref ...
reading the sense bank [sense] .... Enriching parse with sense ...
reading the name bank [name].... Enriching parse with name ...
reading the parallel bank [parallel] ....

finding original trees to prepare for parallel bank enrichment....

Loading banks for all@msnbc@bc@en@on: parse, coref, sense, name, parallel, prop, speaker ...
reading the treebank [parse] .... 660 trees in the treebank
reading the coreference bank [coref] ....Enriching parse with coref ...
reading the sense bank [sense] .... Enriching parse with sense ...
reading the name bank [name].... Enriching parse with name ...
reading the parallel bank [parallel] ....
reading the proposition bank [prop] .... Enriching parse with prop ...
reading the speaker bank [speaker] .... Enriching parse with speaker ...
In [140]: s_0 = o[0]

Loading banks for all@msnbc@bc@ch@on: parse, coref, sense, name, parallel, prop, speaker 
reading the treebank [parse] .... 665 trees in the treebank 
reading the coreference bank [coref] .... Enriching parse with coref 
reading the sense bank [sense] .... Enriching parse with sense 
reading the name bank [name].... Enriching parse with name 
reading the parallel bank [parallel] 

finding original trees to prepare for parallel bank enrichment ....

Loading banks for all@msnbc@bc@en@on: parse, coref, sense, name, parallel, prop, speaker 
reading the treebank [parse] .... 660 trees in the treebank 
reading the coreference bank [coref] .... Enriching parse with coref 
reading the sense bank [sense] .... Enriching parse with sense 
reading the name bank [name].... Enriching parse with name 
reading the parallel bank [parallel] 
reading the proposition bank [prop] .... Enriching parse with prop 
reading the speaker bank [speaker] .... Enriching parse with speaker 

found 1 original treebanks.

enriching treebanks with tree-to-tree parallel data ..... 
reading the proposition bank [prop] .... Enriching parse with prop 
reading the speaker bank [speaker] .... Enriching parse with speaker
If you try to load the next subcorpus you will not see any output because it has already read it automatically.
If you try to load the next subcorpus you will not see any output because it has already read it automatically.

In [141]: s_1 = o[1]
If you try to load the next subcorpus you will not see any output because it has already read it automatically.

In [141]: s_1 = o[1]

In [142]:
But, they are different
But, they are different

In [143]: s_0
But, they are different

In [143]: s_0
Out[143]:
subcorpus instance, id=all@msnbc@bc@ch@on, banks:
  [  coref] : gold@all@msnbc@bc@ch@on
  [document] : gold@all@msnbc@bc@ch@on
  [  name] : gold@all@msnbc@bc@ch@on
  [parallel] : gold@all@msnbc@bc@ch@on
  [parse] : gold@all@msnbc@bc@ch@on
  [  prop] : gold@all@msnbc@bc@ch@on
  [  sense] : gold@all@msnbc@bc@ch@on
  [speaker] : gold@all@msnbc@bc@ch@on
But, they are different

In [143]: s_0
Out[143]:
subcorpus instance, id=all@msnbc@bc@ch@on, banks:
    [  coref] : gold@all@msnbc@bc@ch@on
    [document] : gold@all@msnbc@bc@ch@on
    [ name] : gold@all@msnbc@bc@ch@on
    [parallel] : gold@all@msnbc@bc@ch@on
    [ parse] : gold@all@msnbc@bc@ch@on
    [ prop] : gold@all@msnbc@bc@ch@on
    [ sense] : gold@all@msnbc@bc@ch@on
    [ speaker] : gold@all@msnbc@bc@ch@on

In [144]: s_1
But, they are different

In [143]: s_0
Out[143]:
subcorpus instance, id=all@msnbc@bc@ch@on, banks:
[ coref] : gold@all@msnbc@bc@ch@on
[ document] : gold@all@msnbc@bc@ch@on
[ name] : gold@all@msnbc@bc@ch@on
[ parallel] : gold@all@msnbc@bc@ch@on
[ parse] : gold@all@msnbc@bc@ch@on
[ prop] : gold@all@msnbc@bc@ch@on
[ sense] : gold@all@msnbc@bc@ch@on
[ speaker] : gold@all@msnbc@bc@ch@on

In [144]: s_1
Out[144]:
subcorpus instance, id=all@msnbc@bc@en@on, banks:
[ coref] : gold@all@msnbc@bc@en@on
[ document] : gold@all@msnbc@bc@en@on
[ name] : gold@all@msnbc@bc@en@on
[ parallel] : gold@all@msnbc@bc@en@on
[ parse] : gold@all@msnbc@bc@en@on
[ prop] : gold@all@msnbc@bc@en@on
[ sense] : gold@all@msnbc@bc@en@on
[ speaker] : gold@all@msnbc@bc@en@on
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
**Exploring Parallel Connections**
Advanced Topics
Cross-Layer Query

In [145]:
   ct = sb0["parse"]

In [146]:
e = ct["parse"]

Out[147]:
   treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
      [0] : bc/msnbc/00/msnbc@0000@all@msnbc@bc@ch@on

In [148]:
e = ct["parse"]

Out[148]:
   treebank instance, id=gold@all@msnbc@bc@en@on, documents:
      [0] : bc/msnbc/00/msnbc@0005@all@msnbc@bc@en@on

In [149]:
c = ct[0]

In [150]:
e = ct[0]

In [151]:
c0 = c[0]

In [152]:
e = c[0]
In [145]: c_t_b = s_0["parse"]
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ononotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:

treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
[0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:
treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
[0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on

In [148]: e_t_b
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:
treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
  [0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on

In [148]: e_t_b
Out[148]:
treebank instance, id=gold@all@msnbc@bc@en@on, documents:
  [0] : bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:
treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
   [0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on

In [148]: e_t_b
Out[148]:
treebank instance, id=gold@all@msnbc@bc@en@on, documents:
   [0] : bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on

In [149]: c_t_doc = c_t_b[0]
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:
treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
   [0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on

In [148]: e_t_b
Out[148]:
treebank instance, id=gold@all@msnbc@bc@en@on, documents:
   [0] : bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on

In [149]: c_t_doc = c_t_b[0]
In [150]: e_t_doc = e_t_b[0]
Pradhan, Xue
OntoNotes: The 90% Solution

In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:
treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
[0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on

In [148]: e_t_b
Out[148]:
treebank instance, id=gold@all@msnbc@bc@en@on, documents:
[0] : bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on

In [149]: c_t_doc = c_t_b[0]
In [150]: e_t_doc = e_t_b[0]
In [151]: c_t_0 = c_t_doc[0]
In [145]: c_t_b = s_0["parse"]
In [146]: e_t_b = s_1["parse"]
In [147]: c_t_b
Out[147]:
treebank instance, id=gold@all@msnbc@bc@ch@on, documents:
    [0] : bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on

In [148]: e_t_b
Out[148]:
treebank instance, id=gold@all@msnbc@bc@en@on, documents:
    [0] : bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on

In [149]: c_t_doc = c_t_b[0]
In [150]: e_t_doc = e_t_b[0]
In [151]: c_t_0 = c_t_doc[0]
In [152]: e_t_0 = e_t_doc[0]
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [153]: c_t_0
In [153]: c_t_0
Out[153]:
<on.corpora.tree object id=0@bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on value=<<
(TOP (IP (CODE [speaker1_#1E])
  (IP (NP-SBJ (-NONE- *pro*))
    (VP (NP-PRD (NP (NP-PN (NR ...)))
      (NP-PN (NP-PN (NP-PN (NP-PN (NN ...))))
        (NP (NN ...))
      )
    )
  )
)
(PU ...)
(IP (NP-SBJ (PN ...))
  (VP (VC ...)
    (NP-PRD (DNP (NP-PN (NP-PN (NP-PN (NP-PN (NN ...))))
      (DEG ...))
    )
  )
  (NP-PN (PU ...)
    (IP (NP-SBJ (-NONE- *pro*))
      (VP (VV ...)
        (NP-OBJ (NN ...)))
    )
  )
)
(PU ...)))))

Pradhan, Xue

OntoNotes: The 90% Solution
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [154]: e_t_0
In [154]: e_t_0
Out[154]:
<on.corpora.tree object id=0@bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on value=<
(TOP (S (CODE [speaker1])
 (PP (IN From)
  (NP (NNP ~NBC)
   (NN news))
  (PP-LOC (IN in)
   (NP (NNP Washington))))))
(NP-SBJ (DT this))
(VP (VBZ is)
 (NP-PRD (NP (NNP Meet)
   (NNP the)
    (NNP Press))
  (PP (IN with)
   (NP (NNP Jim)
     (NNP Russert))))))
(././.))>

Pradhan, Xue

OntoNotes: The 90% Solution
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ononote
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [155]: c_t_0.originals
In [155]: c_t_0.originals
Out[155]:
[<on.corpora.tree object id=0@bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on value=<
(TOP (S (CODE [speaker1])
   (PP (IN From)
       (NP (NP (NNP ~NBC)
           (NN news))
         (PP-LOC (IN in)
             (NP (NNP Washington))))))
   (NP-SBJ (DT this))
   (VP (VBZ is)
       (NP-PRD (NP (NNP Meet)
           (NP (NNP the)
             (NNP Press))
           (PP (IN with)
               (NP (NNP Jim)
                 (NNP Russert))))))
   (. /.)>>]
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
In [156]: c_t_0.origina$[0].translations
In [156]: c_t_0.origina ls[0].translations
Out[156]:
[<on.corpora.tree object id=0@bc/msnbc/00/msnbc_0000@all@msnbc@bc@ch@on value=<
(TOP (IP (CODE [speaker1.#1E]))
  (IP (NP-SBJ (-NONE- *pro*)))
    (VP (NP-PRD (NP (NP-PN (NR ...))
      (NP-PN (NP-PN (NN ...)
        (NP-PN (NN ...))
        (NN ...))
        (NP (NN ...))))))
  (PU ...)
  (IP (NP-SBJ (PN ...)))
    (VP (VC ...)
      (NP-PRD (DNP (NP-PN (NP-PN (NR ...))
        (DEG ...))
        (NP-PN (PU ...)
          (IP (NP-SBJ (-NONE- *pro*)))
            (VP (VV ...)
              (NP-OBJ (NN ...))))))
    (PU ...))
  (PU ...)])

Pradhan, Xue
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ononotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

In [167]: len(doc)
Out[167]: 665

In [169]: for c in range(0, len(doc)):
   if(len(doc[c]. originals) > 1):
      print c

643

In [172]: doc[643] =...:

Pradhan, Xue
OntoNotes: The 90% Solution
In [167]: len(c_t_doc)
In [167]: len(c_t_doc)
Out[167]: 665
In [167]: len(c_t_doc)
Out[167]: 665

In [169]: for c_t_index in range(0, len(c_t_doc)):
   if(len(c_t_doc[c_t_index].originals) > 1):
       print c_t_index

       ......:

       ......:
In [167]: len(c_t_doc)
Out[167]: 665

In [169]: for c_t_index in range(0, len(c_t_doc)):
   ...:     if(len(c_t_doc[c_t_index].originals) > 1):
   ...:         print c_t_index
   ....:
   ....:
643
In [167]: len(c_t_doc)
Out[167]: 665

In [169]: for c_t_index in range(0, len(c_t_doc)):
   if(len(c_t_doc[c_t_index].originals) > 1):
       print c_t_index
   .......
   .......
643

In [172]: c_t_643 = c_t_doc[643]
Challenges with Multiple Layers of Annotation
Architecture
Raw Data
Database Design
Python API Design
Data Access

Configuration
Creating ontonotes
Exploring Various Layers
Exploring Parallel Connections
Advanced Topics
Cross-Layer Query

Pradhan, Xue
OntoNotes: The 90% Solution
Pradhan, Xue

In [173]: c_t_643.originals
In [173]: c_t_643.originals
Out[173]:
[<on.corpora.tree object id=638@bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on value=<<
(TOP (S (CODE [Tim Russert])
  (NP-SBJ-1 (PRP I))
  (VP (VBD was)
    (ADJP-PRD (JJ ready)
      (S (NP-SBJ (-NONE- *PRO*-1))
        (VP (TO to)
          (VP (VB wear)
            (NP (DT this))
            (PP-PRP (IN for)
              (NP (DT the)
                (JJ final)
                (CD four)))))),
      (. /-.)))))),
  <on.corpora.tree object id=639@bc/msnbc/00/msnbc_0005@all@msnbc@bc@en@on value=<<
(TOP (S-UNF (CODE [Tim Russert])
  (CC but)
  (INTJ (UH uh))
  (INTJ (UH uh))
  (NP-SBJ (PRP I))
  (. /-.)))]

Pradhan, Xue
OntoNotes: The 90% Solution
Advanced Configuration

```
[corpus]
data_in : data
load : english-nw-wsj
granularity : source
banks : parse coref sense name prop
          b_parse:parse b_sense:b_parse b_prop:b_parse
ignore-inventories: senses frames
```
Dealing with Alignment

- Examples on Live CD
What is the distribution of named entities that are ARG0s of the predicate "say"?

```python
for a_proposition in a_proposition_bank:
    if (a_proposition.lemma == "say"):
        query = "select * from argument where proposition_id = '%s';" % (a_proposition.id)
        a_cursor.execute(query)
        argument_rows = a_cursor.fetchall()
        for a_argument_row in argument_rows:
            a_argument_id = a_argument_row['id']
            a_argument_type = a_argument_row['type']
            if (a_argument_type == "ARG0"):
                n_in_arg_q = "select * from argument_node where argument_id = '%s';" % (a_argument_id)
                a_cursor.execute(n_in_arg_q)
                argument_node_rows = a_cursor.fetchall()
                for a_argument_node_row in argument_node_rows:
                    a_node_id = a_argument_node_row['id']
                    a_ne_node_query = "select * from name_entity where subtree_id = '%s';" % (a_node_id)
                    a_cursor.execute(a_ne_node_query)
                    ne_rows = a_cursor.fetchall()
                    for a_ne_row in ne_rows:
                        a_ne_type = a_ne_row['type']
                        ne_hash[a_ne_type] = ne_hash[a_ne_type] + 1

if (proposition.lemma == "say"):
    for child in node.subtrees():
        ... ...
```

<table>
<thead>
<tr>
<th>Name Entity</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Person</td>
<td>84</td>
</tr>
<tr>
<td>GPE</td>
<td>34</td>
</tr>
<tr>
<td>Organization</td>
<td>29</td>
</tr>
<tr>
<td>NORP</td>
<td>15</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Pradhan, Xue

OntoNotes: The 90% Solution
Acknowledgment