

# OntoNotes: The 90% Solution

Sameer S Pradhan<sup>1</sup> Nianwen Xue<sup>2</sup>

<sup>1</sup>BBN Technologies, Cambridge, MA

<sup>2</sup>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)

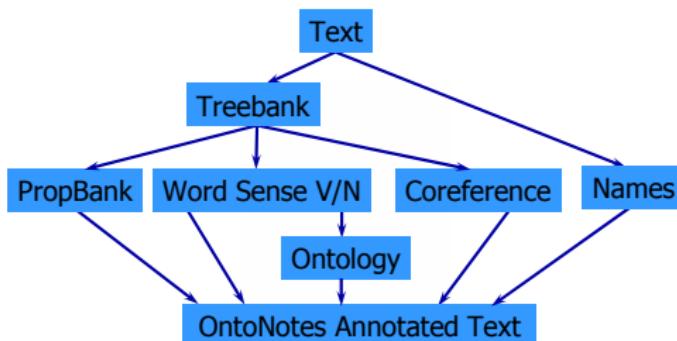
- Multiple layers of annotation

- Syntax
- Propositions
- Word sense
- Coreference
- Names
- Ontology

- Multilingual resource

- English
- Chinese
- Arabic

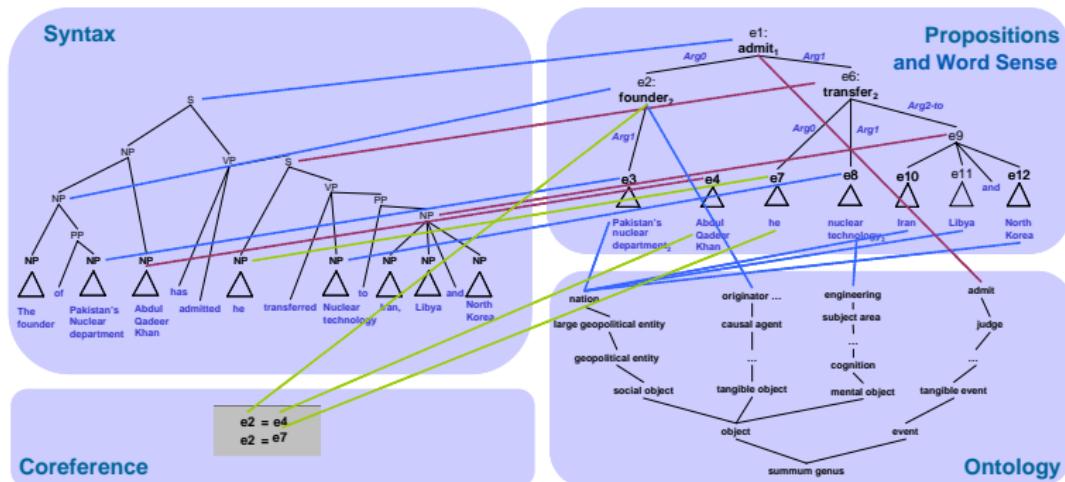
- Parallel Data



## 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 Grand View



The Founder of Pakistan's nuclear department, Abdul Qadeer Khan, has admitted he transferred nuclear technology to Iran, Libya and North Korea.

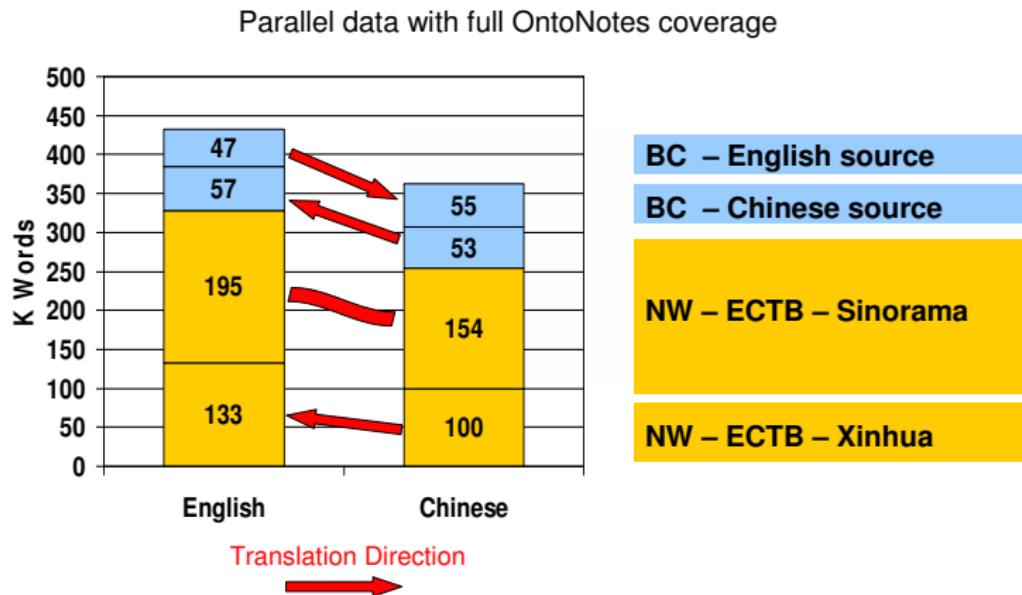
## Amount of Data

### OntoNotes 3.0

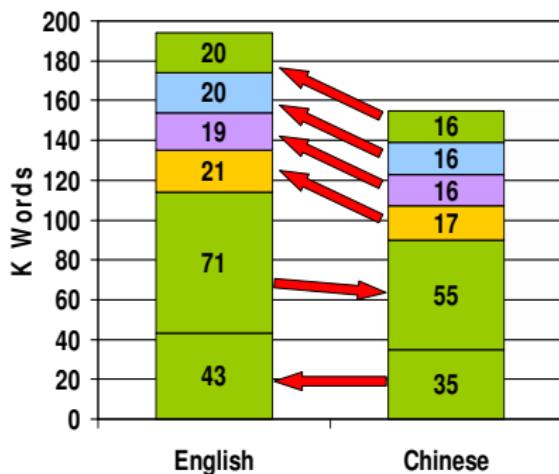
	NW	BN	BC
English	500	200	200
Chinese	250	300	150
Arabic	200		

(K Words)

# Parallel Data Quantities: Full OntoNotes



# Parallel Data Quantities: Only Treebank



# 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

## An Example (Document)

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.

## Example (Names)

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

Token number  
and height in tree

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)

Sentence  
and  
token  
numbers

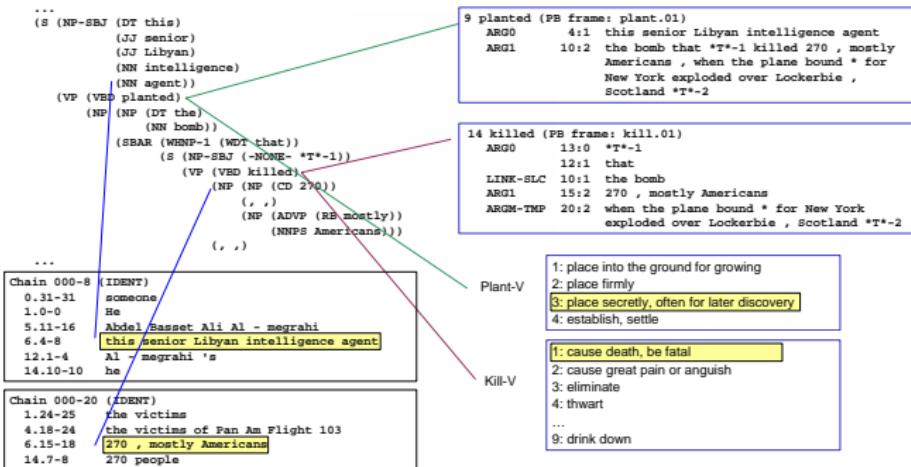
Chain 000-8 (IDENT)  
0.31-31 someone  
1.0-0 He  
5.11-16 Abdel Basset Ali Al - megrahi  
6.4-8 **this senior Libyan intelligence agent**  
12.1-4 Al - megrahi 's  
14.10-10 he

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

# Example (Summary)

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

Annotations	WordNet/ SemCor	Salsa	Prague	OntoNotes
Syntax		✓	✓	✓
Propositions		✓	✓	✓
Sense Tags ITA > 70-80%	✓			✓
Sense Tags ITA > 90%				✓
Coref				✓
> 1M words		✓	✓	✓
Genres		1	1	2+3* *NW, BN, +BC, NG, WebLogs
Languages	1	1	2* *Czech, English,	3* *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

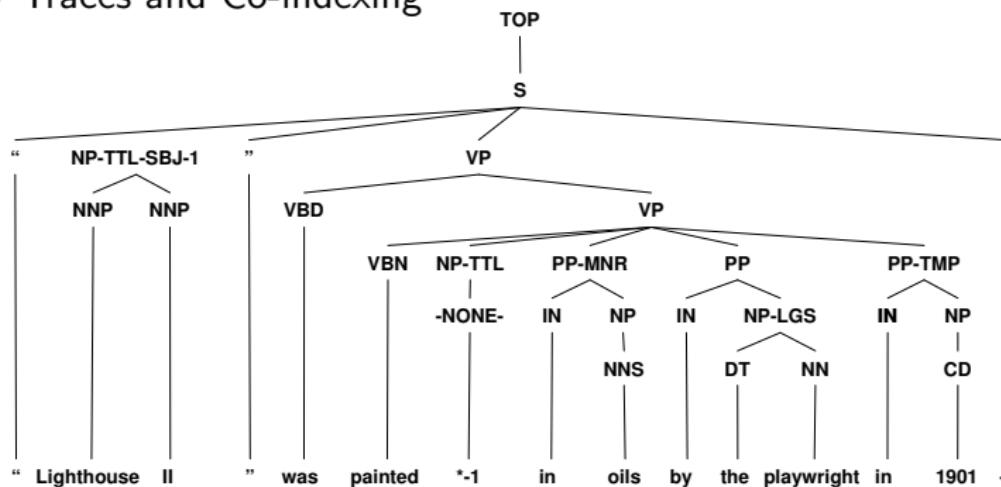
**Treebank**  
PropBank  
Word Sense  
Ontology  
Coreference  
Names

Syntactic Structure  
English Treebank  
Enhancements to the English Treebank  
Chinese Treebank

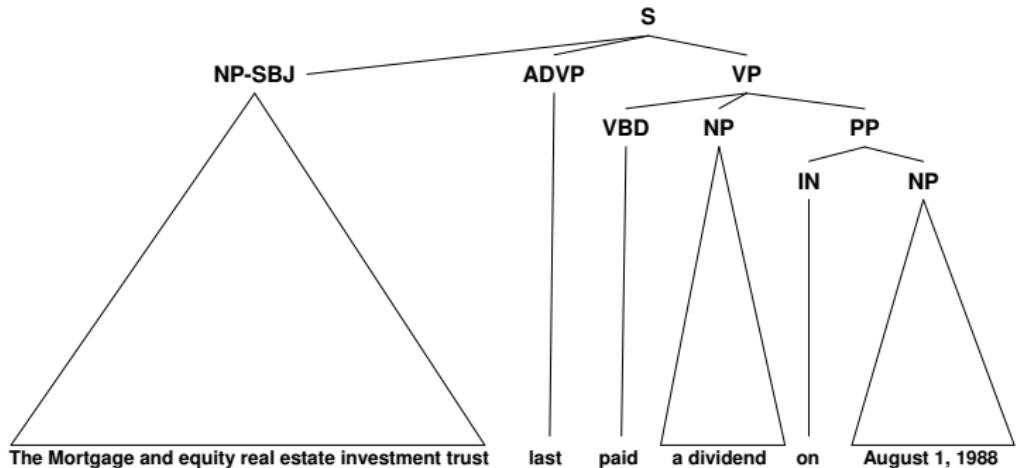
# Treebank

# Syntactic Structure

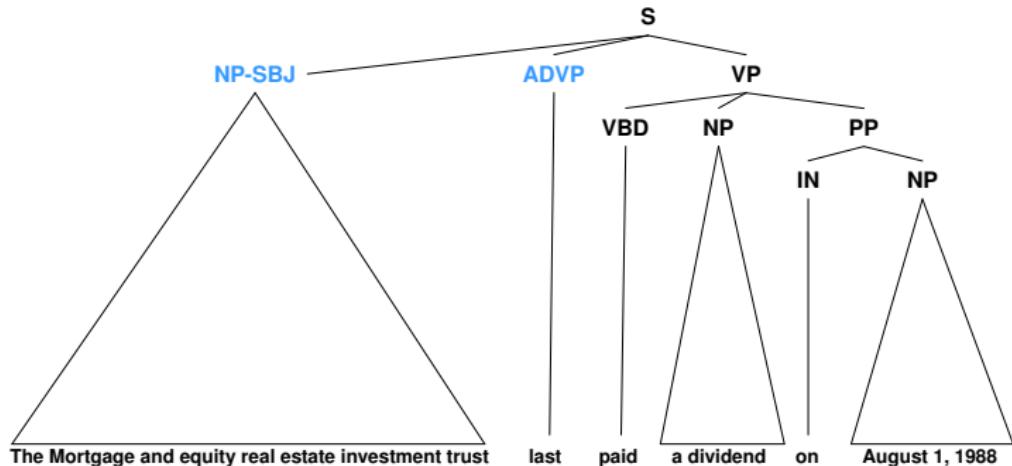
- Phrase Types
- Function Tags
- Traces and Co-indexing



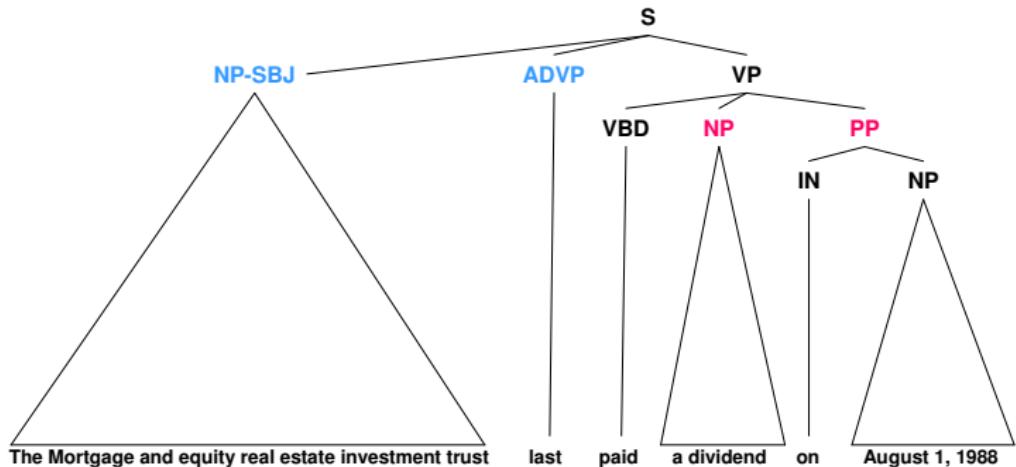
## A English Treebank Example



## A English Treebank Example



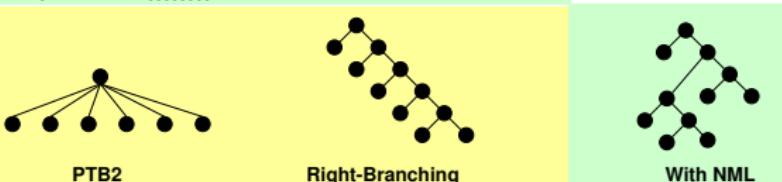
## A English Treebank Example



## 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)))))))



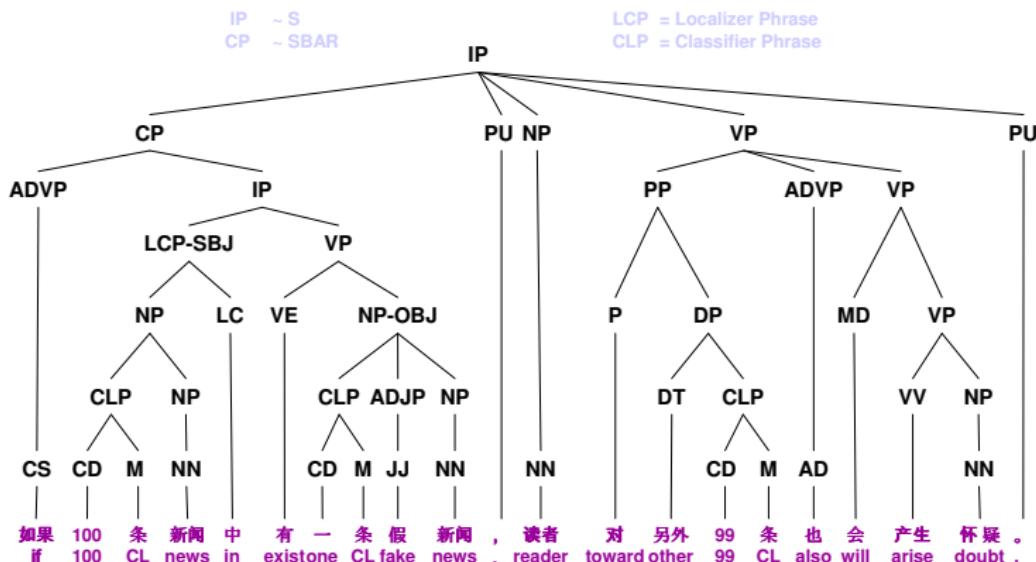
## Improving English Treebank Consistency: Hyphenization

- 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

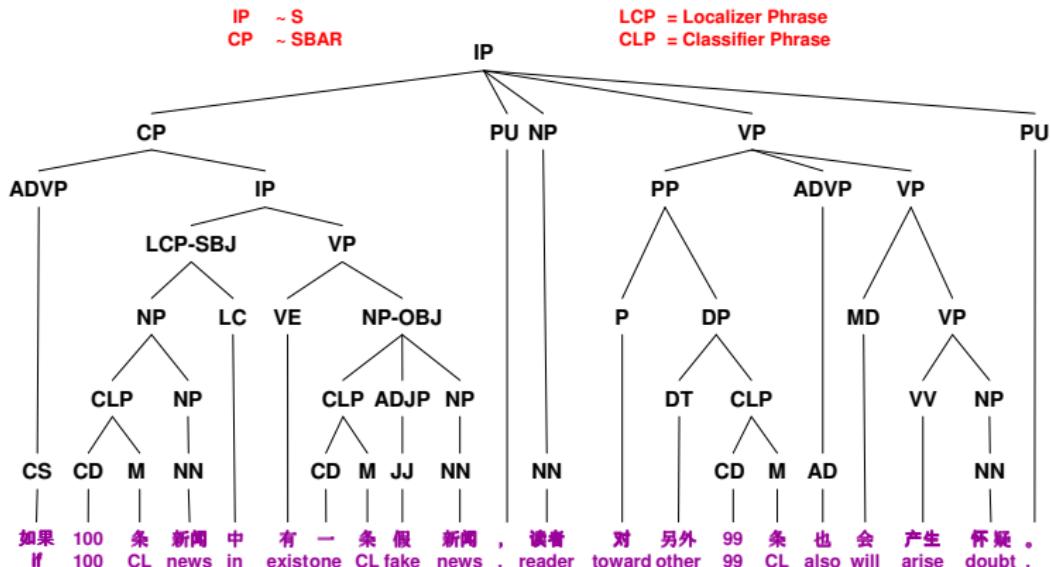
- Penn English Treebank approach
  - Phrase structure annotation
  - Emphasis on trade offs of annotation speed and consistency
  - ITA: 94%
- With enriched structures
  - All structures build on four primitive structures

# A Treebanked Chinese Sentence



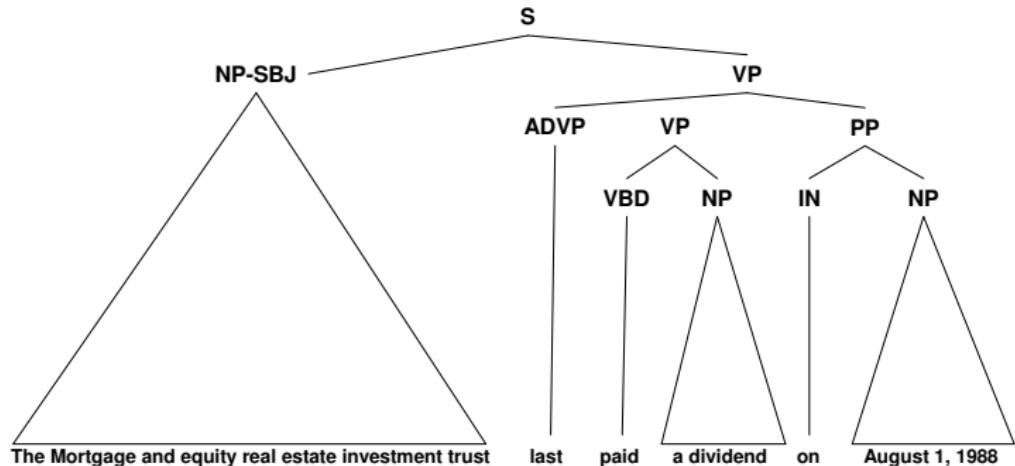
If among the 100 pieces of news one piece is made up, the reader will also doubt the other 99 pieces.

# A Treebanked Chinese Sentence

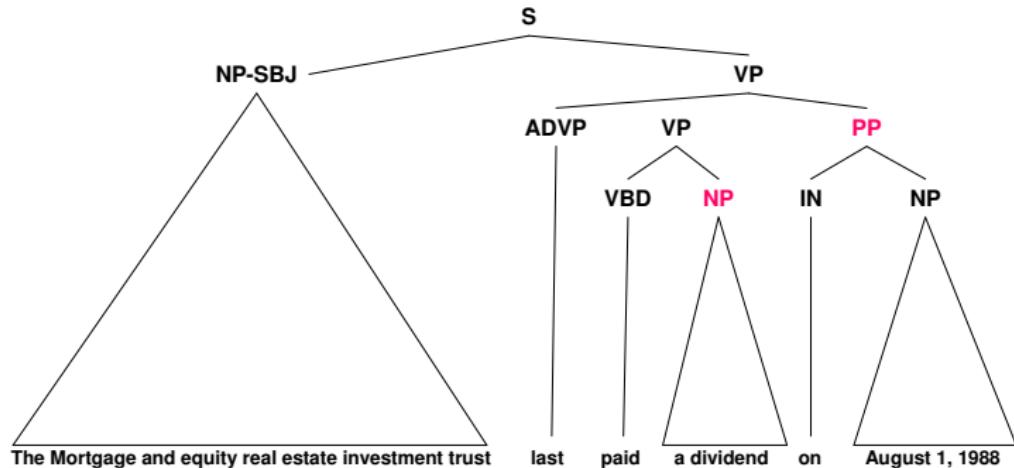


If among the 100 pieces of news one piece is made up, the reader will also doubt the other 99 pieces.

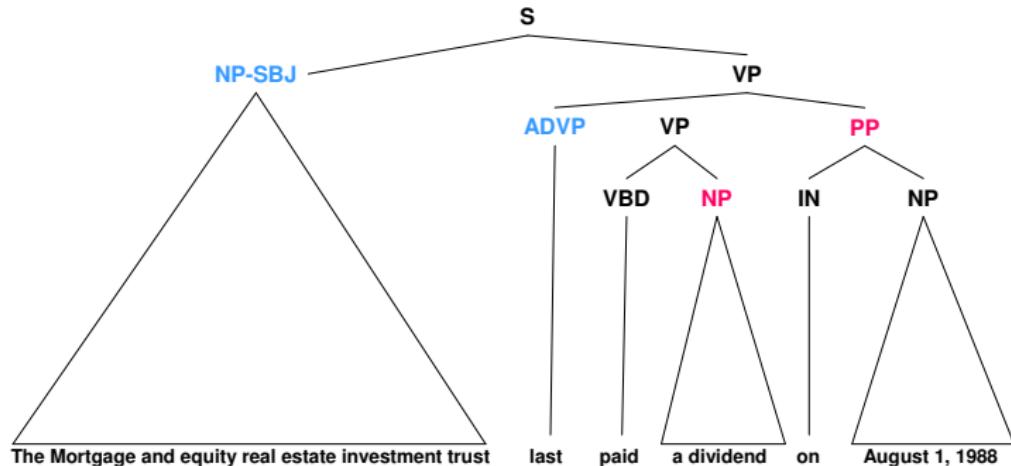
## (Hypothetical) Chinese Treebank



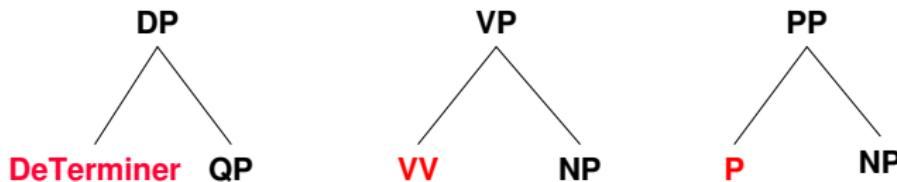
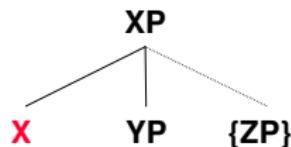
## (Hypothetical) Chinese Treebank



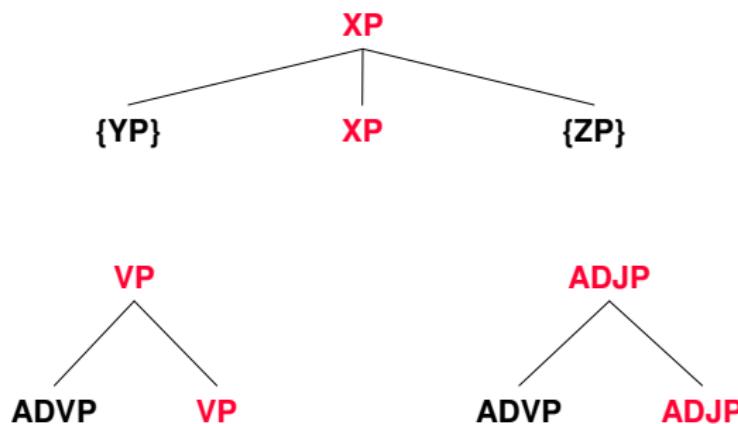
## (Hypothetical) Chinese Treebank



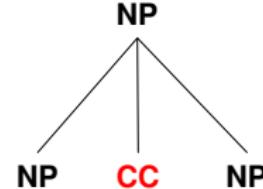
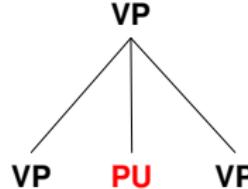
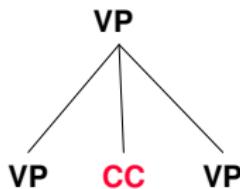
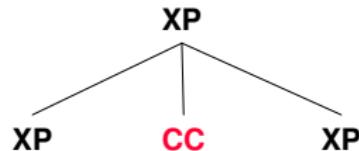
## Complementation (left-headed)



# Adjunction



# Coordination



Treebank  
**PropBank**  
Word Sense  
Ontology  
Coreference  
Names

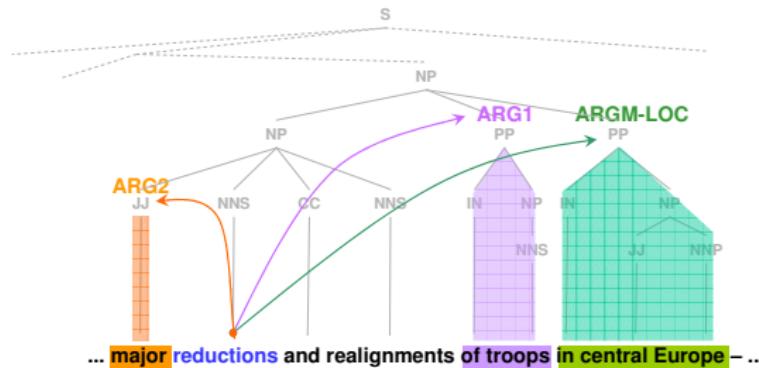
Propositional Structure  
Predicate Frames  
Predicate Senses  
Argument Nomenclature  
Annotation Procedure  
Chinese PropBank  
Treebank and PropBank

# PropBank

# 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

- 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	→	of troops
ARG2 – Amount fallen	→	major
ARG3 – Starting point	-	
ARG4 – Ending point	-	

## 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    | → Net income         |
| ARG2 – amount increased by | → -                  |
| ARG3 – starting point      | → from \$130 million |
| ARG4 – end point           | → To \$274 million   |

## Word Senses in PropBank

- 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

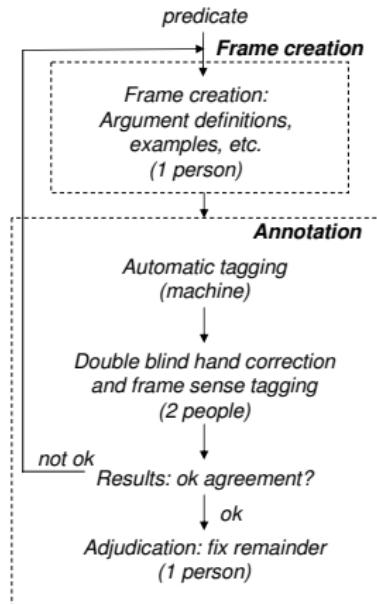
- Arg0 = agent
- Arg1 = direct object/theme/patient
- Arg2 = indirect object/benefactive/instrument/attribute/end state
- Arg3 = start point/benefactive/instrument/attribute
- Arg4 = end point

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

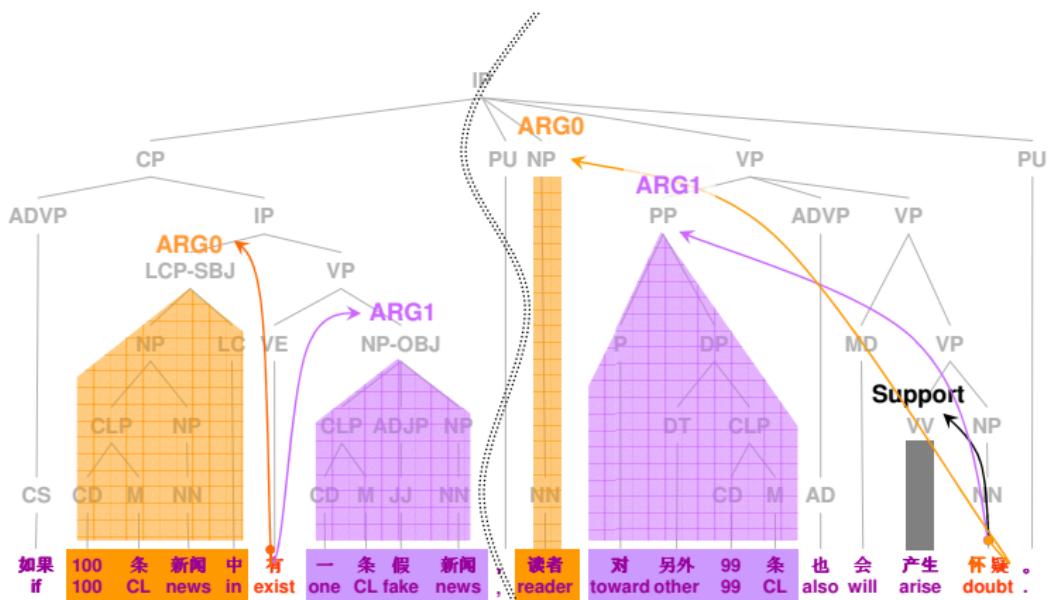
# Annotation Procedure



# Chinese PropBank

- Similar in style to English PropBank
  - Predicate-specific numbered labels for core arguments
  - ArgM<sub>s</sub> for adjunctive arguments
  - Coarse-gained senses
- There are some differences
  - In how split arguments are handled
  - Multi-word expressions are dealt with

# Argument structure of verb and noun



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

# Traces in Chinese PropBank

目前为止，中国 纺织 工业 承建 \*T\* 的 最大 项目  
now till , 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”

## Possession

茅台 酒 制作 工艺 复杂 , 生产 周期 长 。

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”

ARGO-PSR: 茅台酒 “Maotai liquor”

ARGO-PSE: 制作工艺 “brewing process”

REL: 长 “long”

ARGO-PSE: 茅台酒 “Maotai Liquor”

ARGO-PSE: 生产周期 “production cycle”

## Possession (II)

三 大 法 典 须 加 快 出 台 进 程 。

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”

## Predication

西非 经济 明显 恢复 增长。

West African economy clearly resume grow .

“West African economy clearly resumed growing”

---

ARG0: 西非经济 “West African economy”

PRED: 恢复 “resume”

ARGM-ADV: 明显 “clearly”

ARGO-PRD: 增长 “grow”

## Reconciling Treebank and PropBank

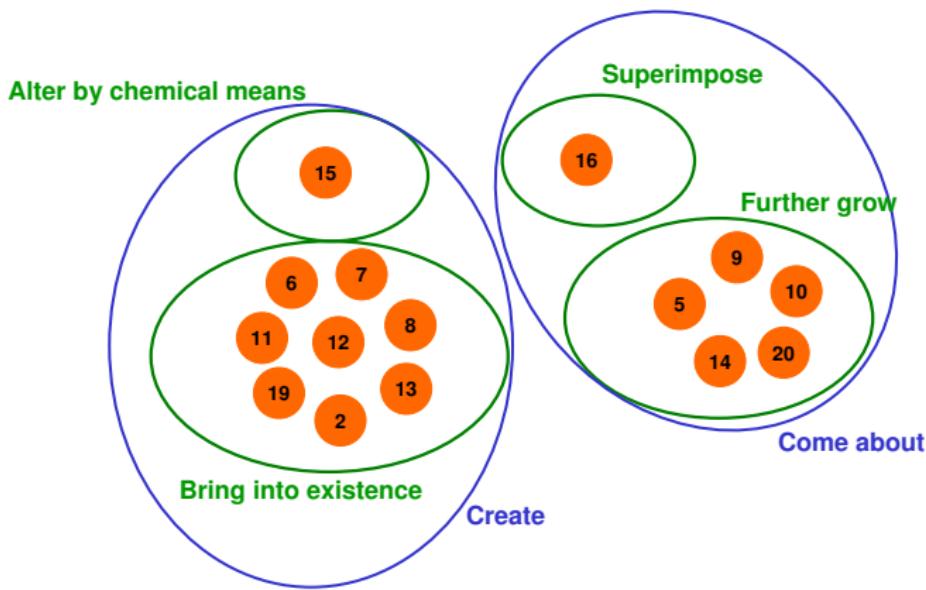
- We found several mismatches between syntax and propositions
  - Sometimes PropBank was right
  - Somethings Treebank was right
- Ambiguities were resolved (PP-attachment)
- Guidelines were modified to bring the two in sync
  - Modified list of verbs that take small-clauses and sentential complements (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

Treebank  
PropBank  
**Word Sense**  
Ontology  
Coreference  
Names

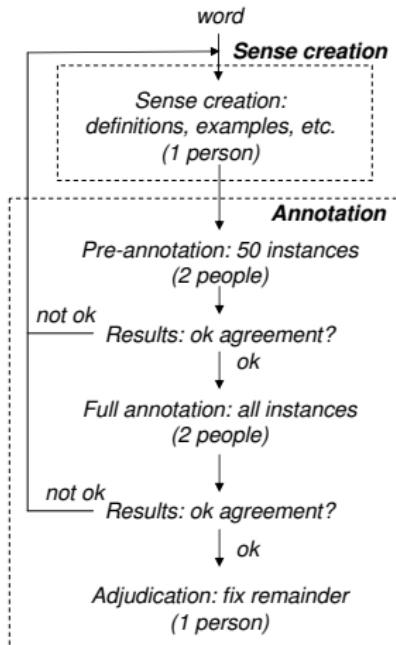
Sense Grouping  
Annotation Procedure  
Connection to Ontology

## Word Sense

## WordNet, OntoNotes and PropBank senses for *develop-v*



## Sense Annotation Procedure



# 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

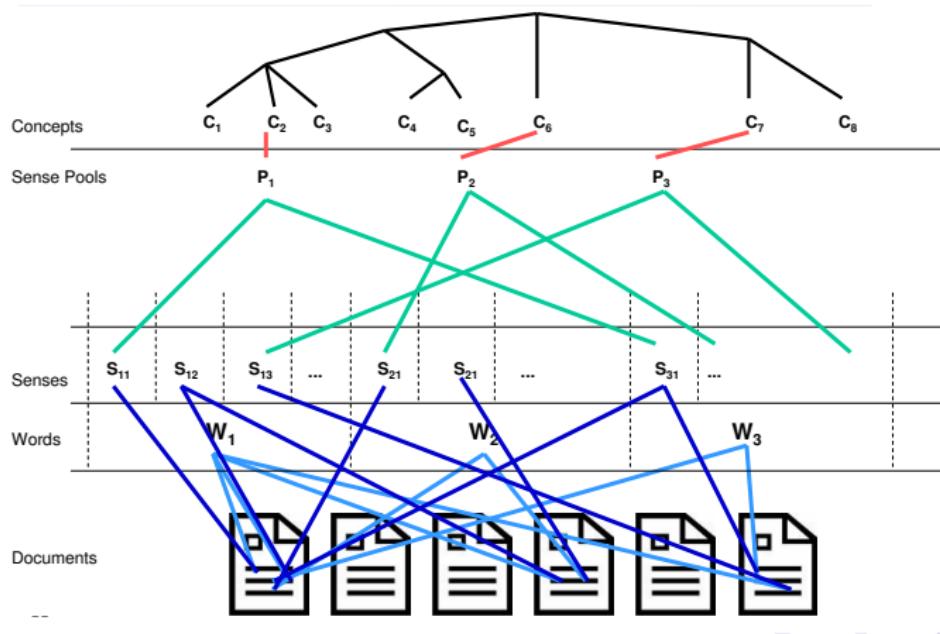
1. Enter into an official record
2. Be aware of, enter into someone's consciousness
3. Indicate a measurement
4. Show in one's face

Treebank  
PropBank  
Word Sense  
**Ontology**  
Coreference  
Names

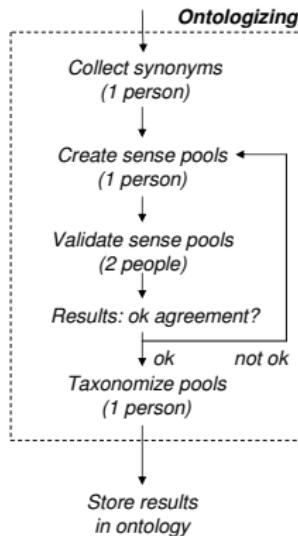
Ontologizing  
Annotation Procedure  
Structure of the Ontology

## Ontology

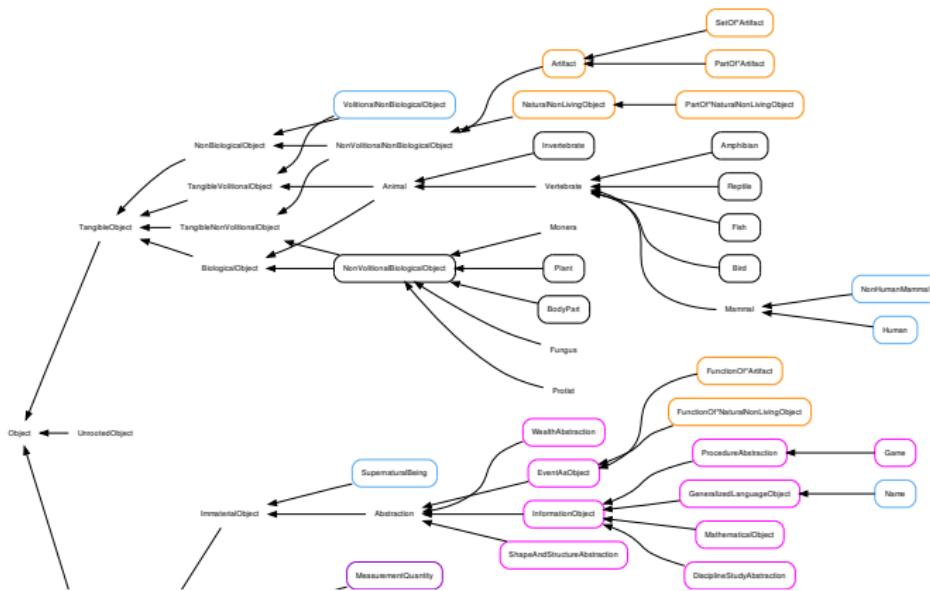
# Ontologizing



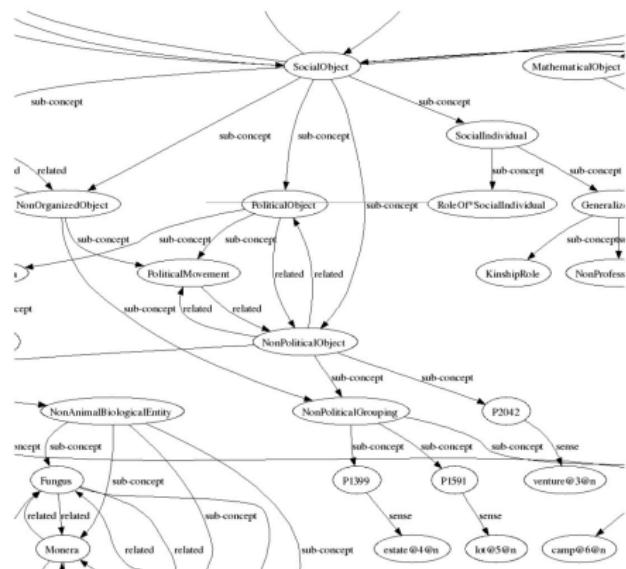
# Sense Pooling Procedure



## Snapshot of the Upper Model



## Snapshot of the Ontology



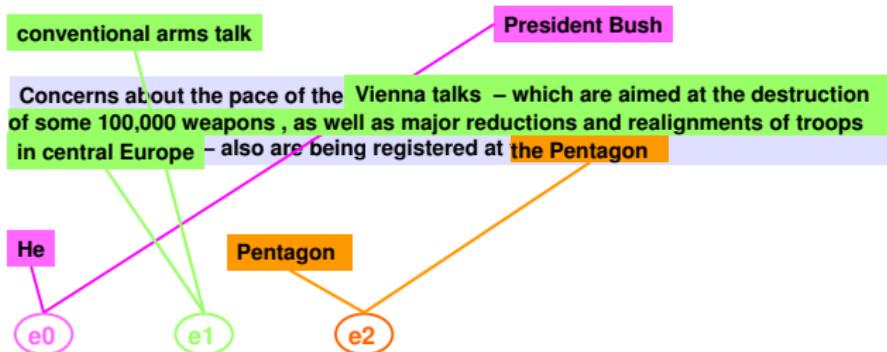
# Ontology Structure

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

## Coreference

# 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)



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

- APPOS

- [[The PhacoFlex intraocular lens]<sub>HEAD</sub>, [the first foldable silicone lens available for cataract surgery]<sub>ATTRIB</sub>]<sub>x</sub>

# 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
- NORG – 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.

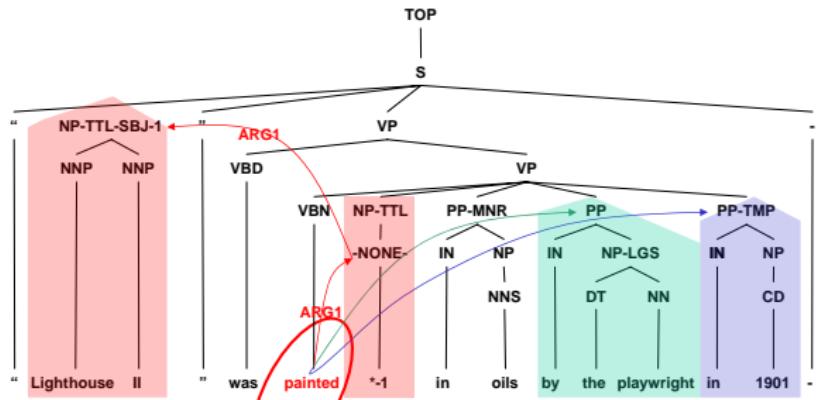
## Types of Names (II)

- 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



```

wsj00/wsj_0037.mrg 67 5 gold set.02 ---- 0:2-ARG0 5:0-rel 6:1-ARG1 10:2-ARGM-TMP
wsj00/wsj_0037.mrg 68 5 gold paint.01 ---- 5:0-rel 1:1*6:0-ARG1 8:1-ARG2-in 10:1-ARG0-by 12:1-ARGM-TMP
wsj00/wsj_0037.mrg 69 21 gold exchange.01 ---- 17:2-ARG0 21:0-rel 22:1-ARG1 23:1-ARGM-TMP
wsj00/wsj_0037.mrg 69 35 gold say.01 ---- 31:1-ARG0 35:0-rel 0:2*37:0-ARG1
  
```

# 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 68 5 paint.01 5:0-rel 1:1'6:0-ARG1 8:1-ARG2-in 10:1-ARG0 12:1-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>
  </frameset>
</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 .

wsj/00/wsj_0037.mrg	0 1	judge-v	2	Sense Number
wsj/00/wsj_0037.mrg	0 36	lot-n	1	

```
<?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

# 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

# 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

## 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

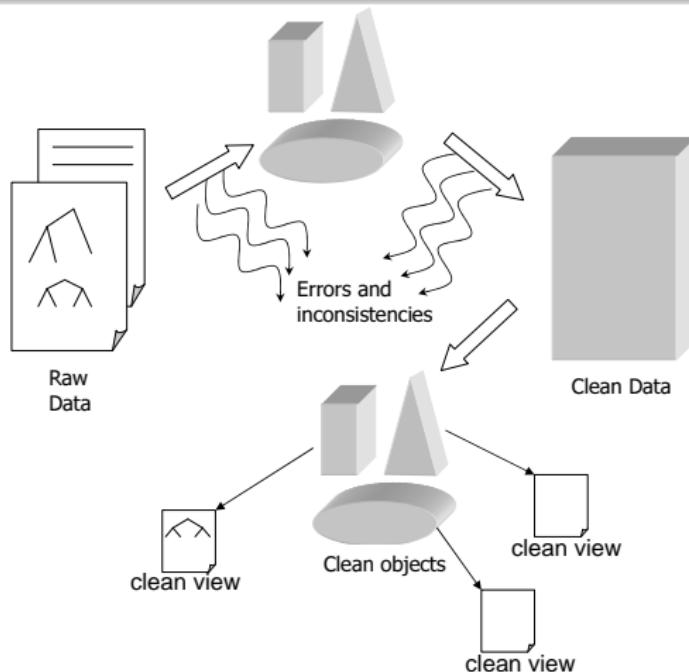
## Advantages of Integrated Representation

- 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

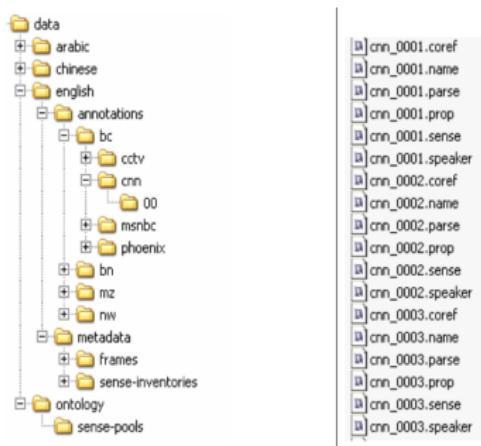
# Data Lifecycle



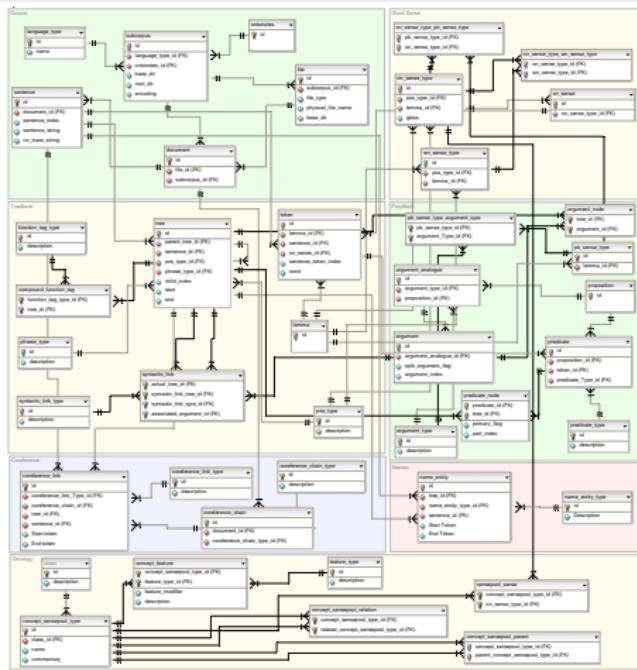
# Organization of the OntoNotes data

```
.../data/<lang>/annotations/<genre>/<source>/<section>/<filename>.<extension>  
.../data/<lang>/metadata/<inventory-type>/<filename>.xml
```

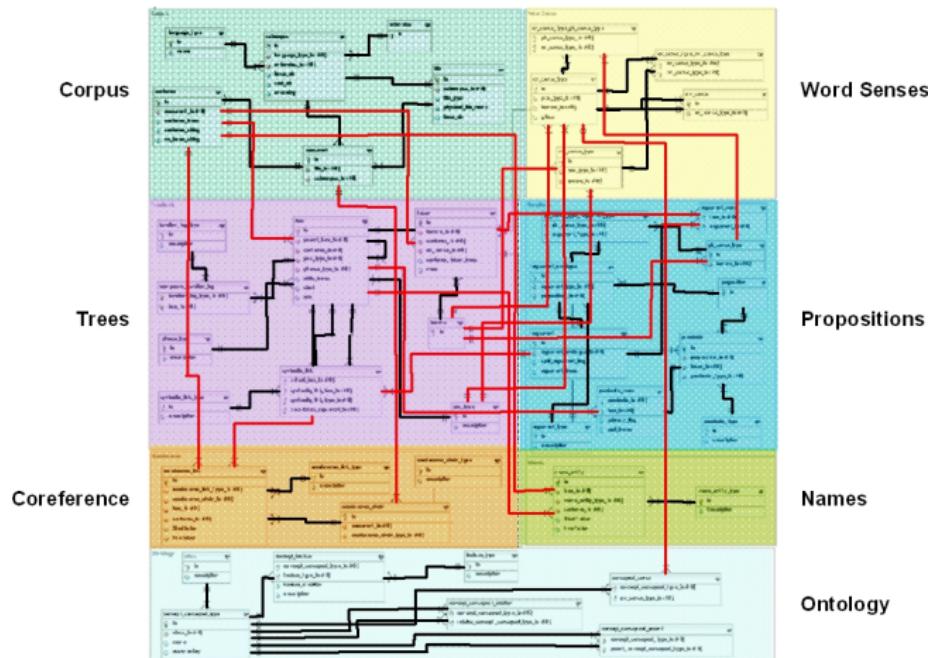
```
<extension> ::= ("parse" | "prop" | "sense" | "coref" | "name" | "parallel" | "speaker")  
<inventory-type> ::= ("frames" | "sense-inventories")
```



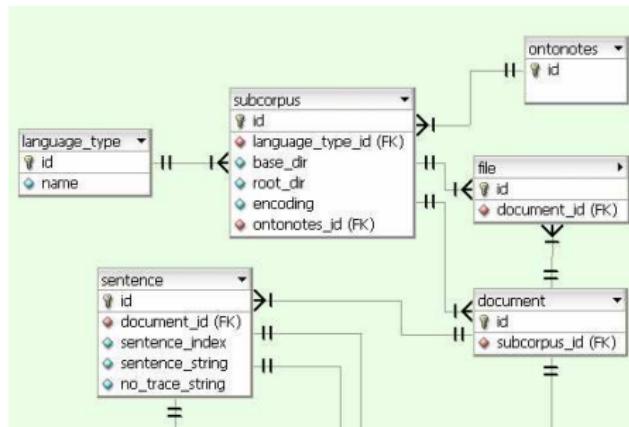
## Entity Relationship Diagram (I)



## Entity Relationship Diagram (II)

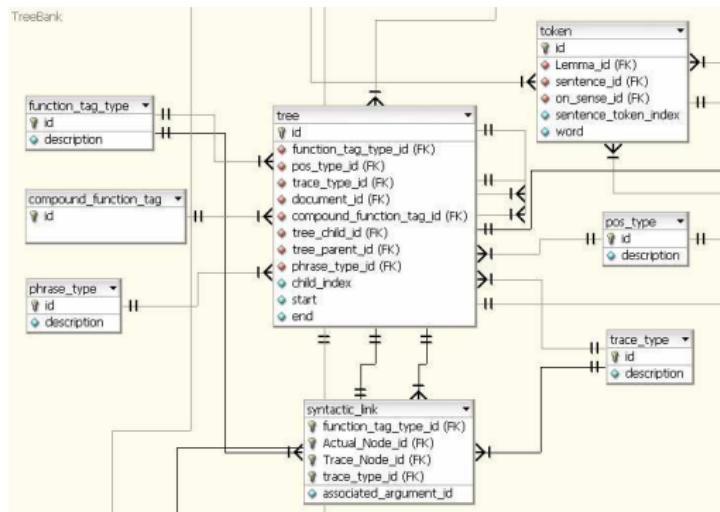


## Corpus Tables



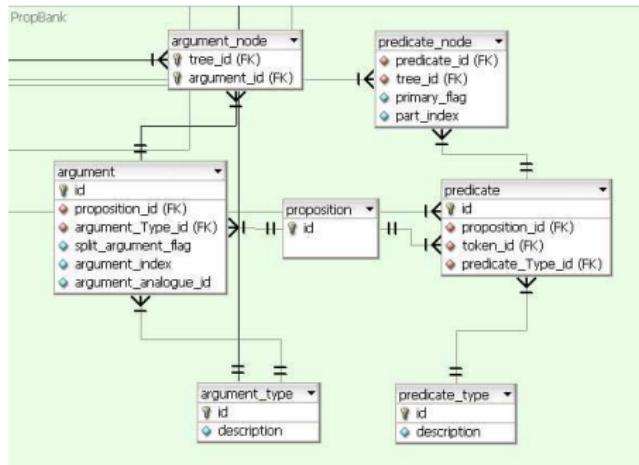
- The corpus tables collectively manage information about the corpus – specifically the subcorpora, documents, files, etc.

# Treebank Tables



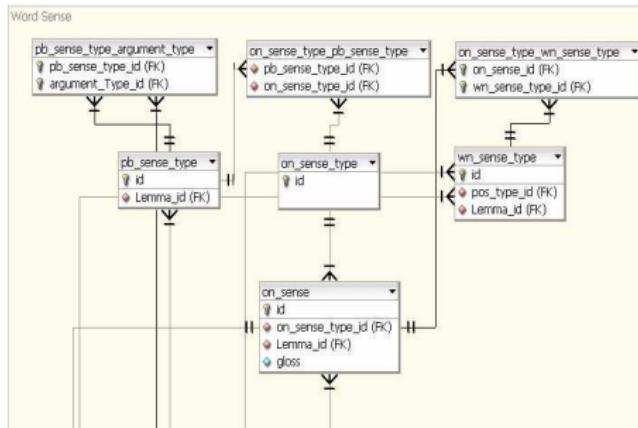
- The treebank tables manage the syntactic tree information. Tokens form the lowest level of granularity in OntoNotes.

# PropBank Tables



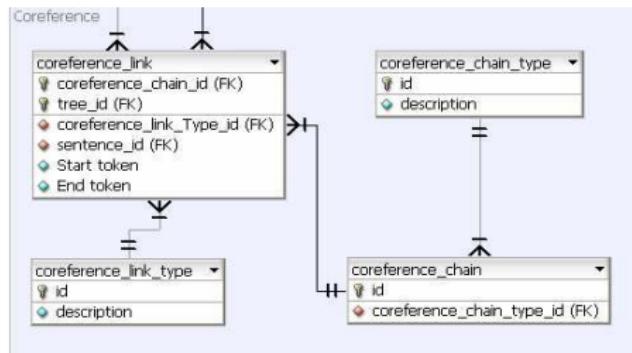
- The proposition tables manage the propositions. The argument\_node forms a composite table to manage many-to-many argument/node relationships

## Word Sense Tables



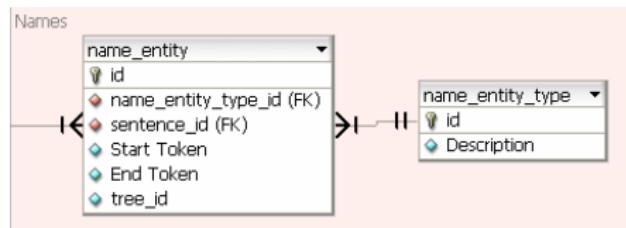
- The sense tables contain the lemma and sense number representing its sense
- Multiple composite tables are used to map WordNet sense, OntoNotes sense and Frame senses to each other

## Coreference Tables



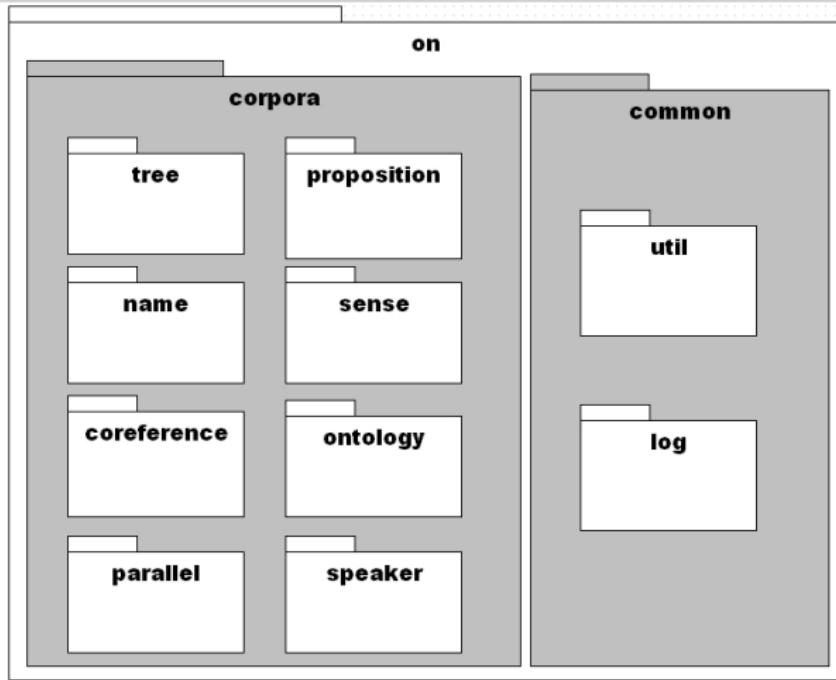
- The **coreference\_chain** and **coreference\_link** tables store the respective pointers.

## Name Tables

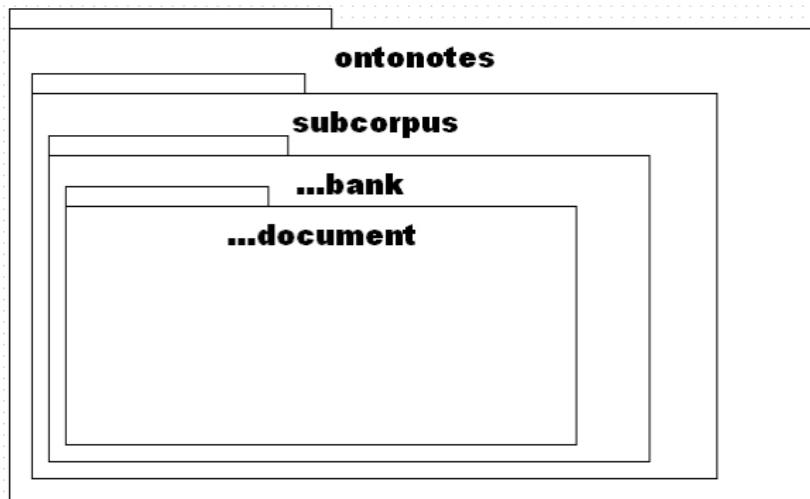


- The name\_entity and name\_type tables represent the names in the corpus

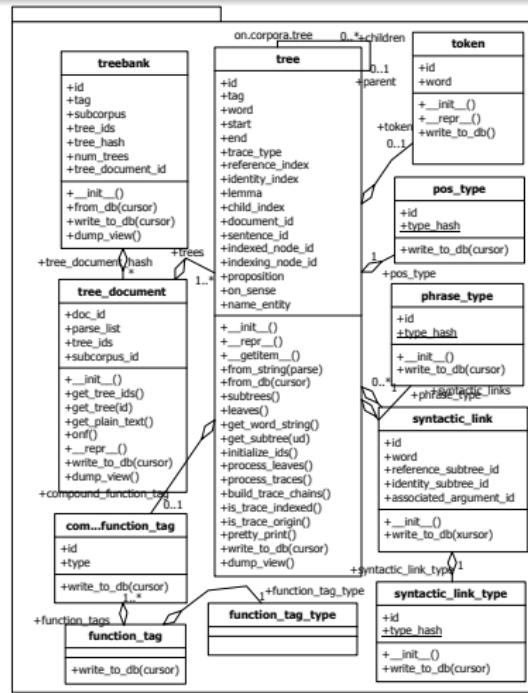
## Module Organization



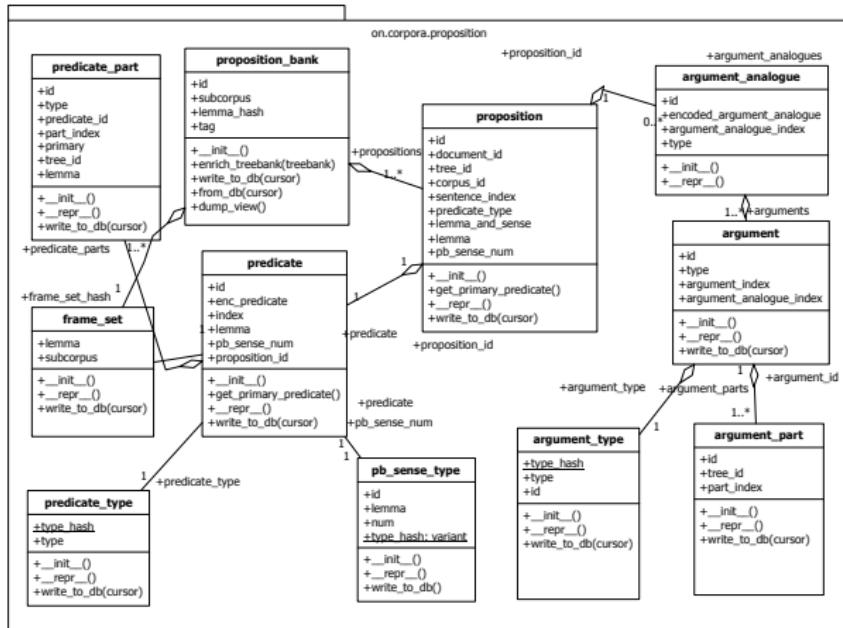
# Object Composition



## on.corpora.tree



# on.corpora.proposition



## Salient Methods

- 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

# Banks

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

Bank Name	Database Table	Python Module	Extention
tree	tree	on.corpora.tree	.parse
sense	on_sense	on.corpora.sense	.sense
proposition	argument, predicate	on.corpora.proposition	.prop
coreference	coreference_link	on.corpora.coreference	.coref
name	name_entity	on.corpora.name	.name
speaker	speaker_sentence	on.corpora.speaker	.speaker
parallel	parallel_sentence, parallel_document	on.corpora.parallel	.parallel

# Treebank

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

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

# PropBank

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

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



# Word Sense

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

Database Tables	Python Objects	File Elements
sense_bank	sense_bank	All .sense files in a <code>on.corpora.subcorpus</code>
None	<code>sense_tagged_document</code>	A single .sense file
on_sense	on_sense	A line in a .sense file
None	<code>sense_inventory</code>	A sense inventory xml file (SI)
on_sense_type	on_sense_type	Fields four and six of a sense line and the <code>inventory/sense</code> element of a SI
on_sense_lemma_type	on_sense_lemma_type	The <code>inventory/its</code> element of a SI
wn_sense_type	wn_sense_type	The <code>inventory/sense/mappings/wn</code> element of a SI
pb_sense_type	pb_sense_type	The <code>inventory/sense/mappings/pb</code> element of a SI
tree	<code>on.corpora.tree.tree</code>	The first three fields of a sense line

# Coreference

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

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

## Name

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

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

# Inventories

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

Database Tables	Python Objects	File Elements
sense_bank	sense_bank	All .sense files in a <code>on.corpora.subcorpus</code>
None	sense_tagged_document	A single .sense file
on_sense	on_sense	A line in a .sense file
None	sense_inventory	A sense inventory xml file (SI)
on_sense_type	on_sense_type	Fields four and six of a sense line and the inventory/sense element of a SI
on_sense_lemma_type	on_sense_lemma_type	The inventory/ita element of a SI
wn_sense_type	wn_sense_type	The inventory/sense/mappings/wn element of a SI
pb_sense_type	pb_sense_type	The inventory/sense/mappings/pb element of a SI
tree	on.corpora.tree.tree	The first three fields of a sense line

## Parallel

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

Database Tables	Python Objects	File Elements
None	<code>parallel_bank</code>	All <code>.parallel</code> files in an <code>on.corpora.subcorpus</code>
<code>parallel_document</code>	<code>parallel_document</code>	The second line (original/translation line) in a <code>.parallel</code> file
<code>parallel_sentence</code>	<code>parallel_sentence</code>	All lines in a <code>.parallel</code> file after the first two (map lines)

# Speaker

DB Tables  $\Leftrightarrow$  Python Objects  $\Leftrightarrow$  File Elements

Database Tables	Python Objects	File Elements
None	<code>speaker_bank</code>	All <code>.speaker</code> files in an <code>on.corpora.subcorpus</code>
None	<code>speaker_document</code>	A <code>.speaker</code> file
<code>speaker_sentence</code>	<code>speaker_sentence</code>	A line in a <code>.speaker</code> file

# 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]

# 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

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

# A Sample Configuration

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

# A Sample Configuration

```
[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
```

# 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
```

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

# Reading the Configuration

In [1]: import on

# 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>
```

# 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'
```

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

# Creating the ontonotes Object

# Creating the ontonotes Object

```
In [9]: o = 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]:
ontonotes 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

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

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

```
In [8]: o = on.ontonotes(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
```

```
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.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
```

```
Out[14]:
```

```
ontonotes instance, id=on, subcorpora:
```

```
[0] : 0089@wsj@nw@en@on
```

```
[1] : 0020@wsj@nw@en@on
```

```
[2] : 0049@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']
```

## 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']
```

```
reading the proposition bank [prop] .....
```

```
Enriching parse with prop ...
```

```
....
```

```
reading the speaker bank [speaker] ...keys: ['parse', 'prop', 'coref', 'name', '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']
```

```
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

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

# Inside the subcorpus

# 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

```
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]
```

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

# Exploring Coreference Data

# 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/wsjs_0020@all@wsj@nw@en@on  
[1] : nw/wsj/00/wsjs_0037@all@wsj@nw@en@on  
[2] : nw/wsj/00/wsjs_0049@all@wsj@nw@en@on  
[3] : nw/wsj/00/wsjs_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

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
```

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 [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@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 [21]: c_chain = c_doc[0]
```

```
In [22]: c_chain
```

```
Out[22]:
```

```
coreference chain instance, id=APPOS@000-52@000@nw/wsj/00/ws_j_0020@all@wsj@nw@en@on, links:  
[0] : ATTRIB@1:2:4@APPOS@000-52@000@nw/wsj/00/ws_j_0020@all@wsj@nw@en@on  
[1] : HEAD@1:6:14@APPOS@000-52@000@nw/wsj/00/ws_j_0020@all@wsj@nw@en@on
```

```
In [24]: c_link_0 = c_chain[0]
```

```
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@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 [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
```

```
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\_chain

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 [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 [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]

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

```
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 [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

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

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.__setattro_
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_sentence_index
c_link_0.get_start_leaf
c_link_0.get_string
c_link_0.get_subtree_id
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.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.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.sentence_index
c_link_0.sql_create_statement
c_link_0.sql_insert_statement
c_link_0.valid
c_link_0.write_to_db
```

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.__setattro_
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_sentence_index
c_link_0.get_start_leaf
c_link_0.get_string
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.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.sentence_index
c_link_0.sql_create_statement
c_link_0.sql_insert_statement
c_link_0.valid
c_link_0.write_to_db
```

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

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'

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

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'

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

# 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
```

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 [52]: t\_0 = t\_doc[0]

In [52]: t\_0 = t\_doc[0]

In [56]: t\_0

In [52]: t\_0 = t\_doc[0]

In [56]: t\_0

Out[56]:

```
<on.corpora.tree object id=0@nw/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)
```

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 [65]: for leaf in t_0.leaves():
....:     print leaf
....:
....:
```

```
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)
(PRP$ 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

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))))))
```

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]

In [19]: t\_0[3:11]

Out[19]:

```
(S-ADV (NP-SBJ (-NONE- *PRO*-1))
       (VP (VBD 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),
 (VBD claiming),
 (DT some),
 (NN success),
 (IN in),
 (PRP$ its),
 (NN trade)]
```

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

# Exploring Proposition Data

# Exploring Proposition Data

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

# Exploring Proposition Data

```
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

In [95]: p[3]

In [95]: p[3]

Out[95]:

proposition:

```
id      : 29@nw/ws_j/00/ws_j_0020@all@ws_j@nw@en@on
doc_id  : nw/ws_j/00/ws_j_0020@all@ws_j@nw@en@on
tree_id : 0@nw/ws_j/00/ws_j_0020@all@ws_j@nw@en@on
frame   : watch.01
enc_prop: ws_j_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/ws_j/00/ws_j_0020@...; enc_self: '29:0-rel'
  < predicate : id: 0@watch.01@v@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '29:0'
    < predicate_node : id: 0@0@watch.01@v@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '29:0'>>>
arguments:
< argument_analogue : id: 0@ARG0@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '26:1-ARG0'
  < argument : id: 0@0@ARG0@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '26:1'
    < argument_node : id: 0@0@0@ARG0@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '26:1'>>>
...
...
links:
< link_analogue : id: 0@LINK-SLC@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '24:1*25:1-LINK-SLC'
  < link : id: 0@0@LINK-SLC@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '24:1'
    < link_node : id: 0@0@0@LINK-SLC@29@0@nw/ws_j/00/ws_j_0020@...; enc_self: '24:1'>>,
```

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

```
In [99]: predicate.[tab]
```

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

In [99]: predicate.[tab]

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

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

In [99]: predicate.[tab]

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

In [99]: predicate.lemma

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

In [99]: predicate.[tab]

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

In [99]: predicate.lemma

Out[99]: u'watch'

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

In [99]: predicate.[tab]

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

In [99]: predicate.lemma

Out[99]: u'watch'

In [100]: predicate.tree\_id

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

In [99]: predicate.[tab]

predicate.__class__	predicate.add	predicate.get_primary_predicate
predicate.__delattr__	predicate.analogue_type	predicate.id
predicate.__dict__	predicate.children	predicate.index_in_parent
predicate.__doc__	predicate.copy_to_different_trees	predicate.lemma
predicate.__getattribute__	predicate.document_id	predicate.parent
predicate.__getitem__	predicate.enc_self	predicate.pb_sense_num
predicate.__hash__	predicate.enc_self_type	predicate.primary_predicate
predicate.__init__	predicate.enrich_tree	predicate.proposition
predicate.__len__	predicate.get_index_of	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 [98]: predicate = p[3].predicate

In [99]: predicate.[tab]

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

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

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
```

```
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 [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

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: 0@0@0@LINK-SLC@29@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: 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
```

```
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: 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 [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@..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 [123]: link_node.subtree
Out[123]: (NP (NNS countries))
```

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 [125]: link\_node.subtree.start

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

```
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/wsj_0020@all@wsj@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'0@0@0@LINK-SLC@29@0@nw/wsj/00/ws_j_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@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 [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'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()  
Out[134]: u'countries'
```

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

# Exploring Senses

# 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/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 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/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
```

# 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]
```

```
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_s0020@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

# Exploring Parallel Connections

# Exploring Parallel Connections

```
In [136]: s["parallel"]
```

# Exploring Parallel Connections

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

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 [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

```
In [139]: o
Out[139]:
ontonotes instance, id=on, subcorpora:
[0] : all@msnbc@bc@ch@on
[1] : 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 [140]: s\_0 = o[0]

```
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@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....
```

```
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]: c\_t\_b = s\_0["parse"]

```
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]
```

```
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

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 [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 (NR ...))
                               (NP-PN (NN ...))
                               (NN ...))
                               (NN ...))))
              (NP (NN ...))))))
          (PU ...))
          (IP (NP-SBJ (PN ...))
              (VP (VC ...)
                  (NP-PRD (DNP (NP-PN (NR ...))
                                 (DEG ...))
                  (NP-PN (PU ...))
                  (IP (NP-SBJ (-NONE- *pro*))
                      (VP (VV ...)
                          (NP-OBJ (NN ...)))))))
              (PU ...))))))
          (PU ...)))>
```

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 [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 (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))))))
  (. /)))>
```

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 [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)  
            (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

```
In [156]: c_t_0.originals[0].translations
```

In [156]: c\_t\_0.originals[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 (NR ...))  
            (NP-PN (NN ...))  
            (NN ...))  
            (NN ...)))  
        (NP (NN ...))))  
    (PU ...)  
    (IP (NP-SBJ (PN ...))  
        (VP (VC ...)  
            (NP-PRD (DNP (NP-PN (NR ...))  
                (DEG ...))  
            (NP-PN (PU ...))  
                (IP (NP-SBJ (-NONE- *pro*))  
                    (VP (VV ...))  
                    (NP-OBJ (NN ...))))  
            (PU ...))))  
    (PU ...))))>]
```

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 [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

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))
        (. /-)))>]
```

# 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

# Cross-Layer Query

What is the distribution of named entities that are ARG0s of the predicate “say”?

```

for a_proposition in a_proposition_bank:
    if proposition_type == "ARG0":
        arg_in_p_q = "select * from argument where proposition_id = '%s';" % (a_proposition.id)
        a_cursor.execute(arg_in_p_q)
        argument_rows = a_cursor.fetchall()
        argument_rows_count = len(argument_rows)

        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 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["node_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

                    a_tree = a_tree_document.get_tree(a_tree_id)
                    a_node = a_tree.get_subtree(a_node_id)

                    for a_child in a_node.subtrees():
                        a_ne_subtree_query = "select * from name_entity where subtree_id = '%s';" % (a_child.id)
                        subtree_ne_rows = a_cursor.execute(a_ne_subtree_query)
                        ne_subtree_rows_count = len(subtree_ne_rows)

                        for a_ne_subtree_row in ne_subtree_rows:
                            a_ne_subtree_type = a_ne_subtree_row["type"]
                            ne_hash[a_subtree_ne_type] = ne_hash[a_subtree_ne_type] + 1

```

**if (proposition.lemma == "say"):**

**query = “select \* from argument where proposition\_id = ‘%s’;” ..**

**if (argument\_type == "ARG0"):**

**for child in node.subtrees():**

Name Entity	Frequency
Person	84
GPE	34
Organization	29
NORP	15
...	...

## Acknowledgment

