LECTURE 2

Annotation Model & Specification

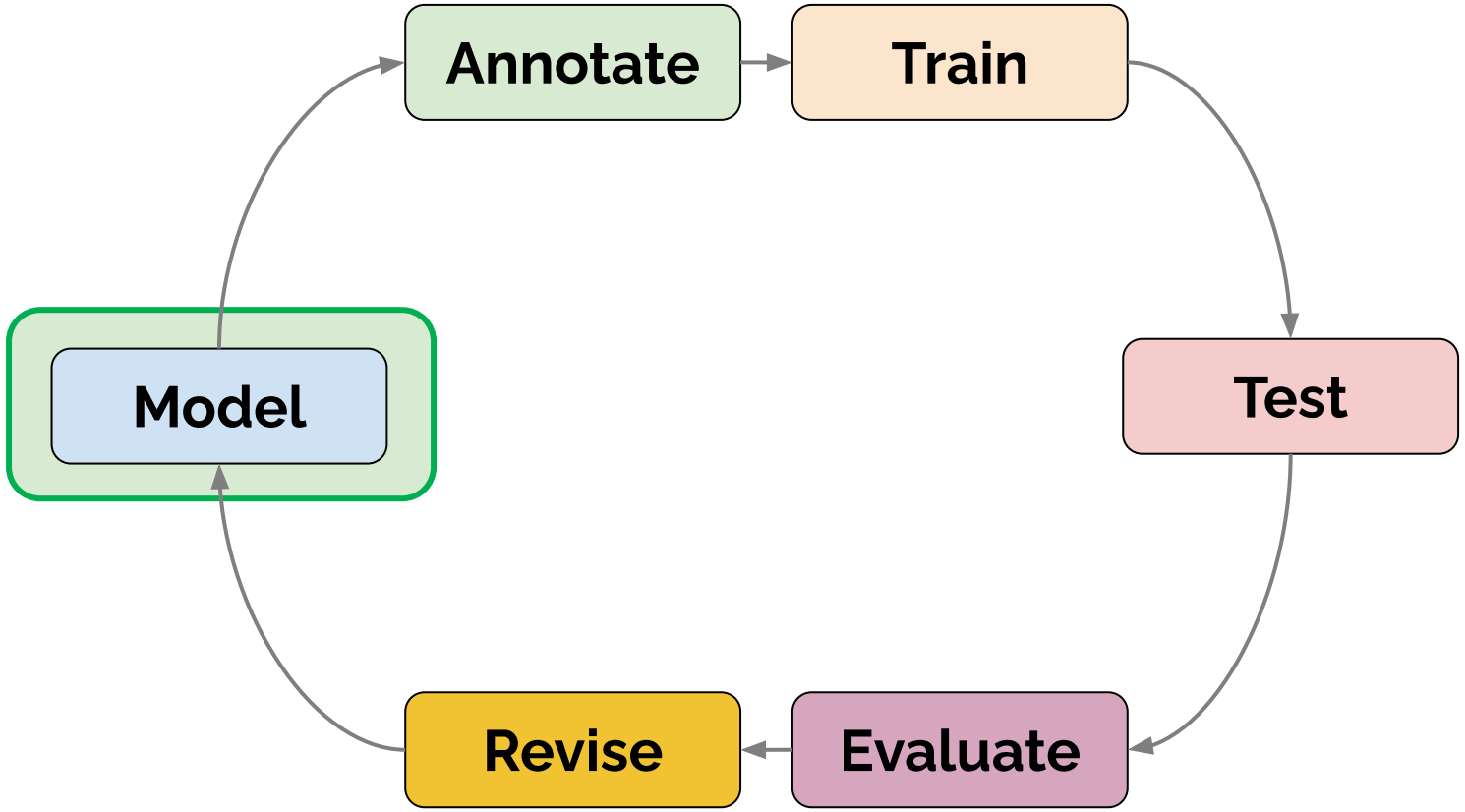
Model & Specification

Annotation standards

Learning Objectives

- Understand how and why to model an annotation process through standardised methods.
- Be aware of best practices in annotation specification.
- Implement an annotation specification in a group.

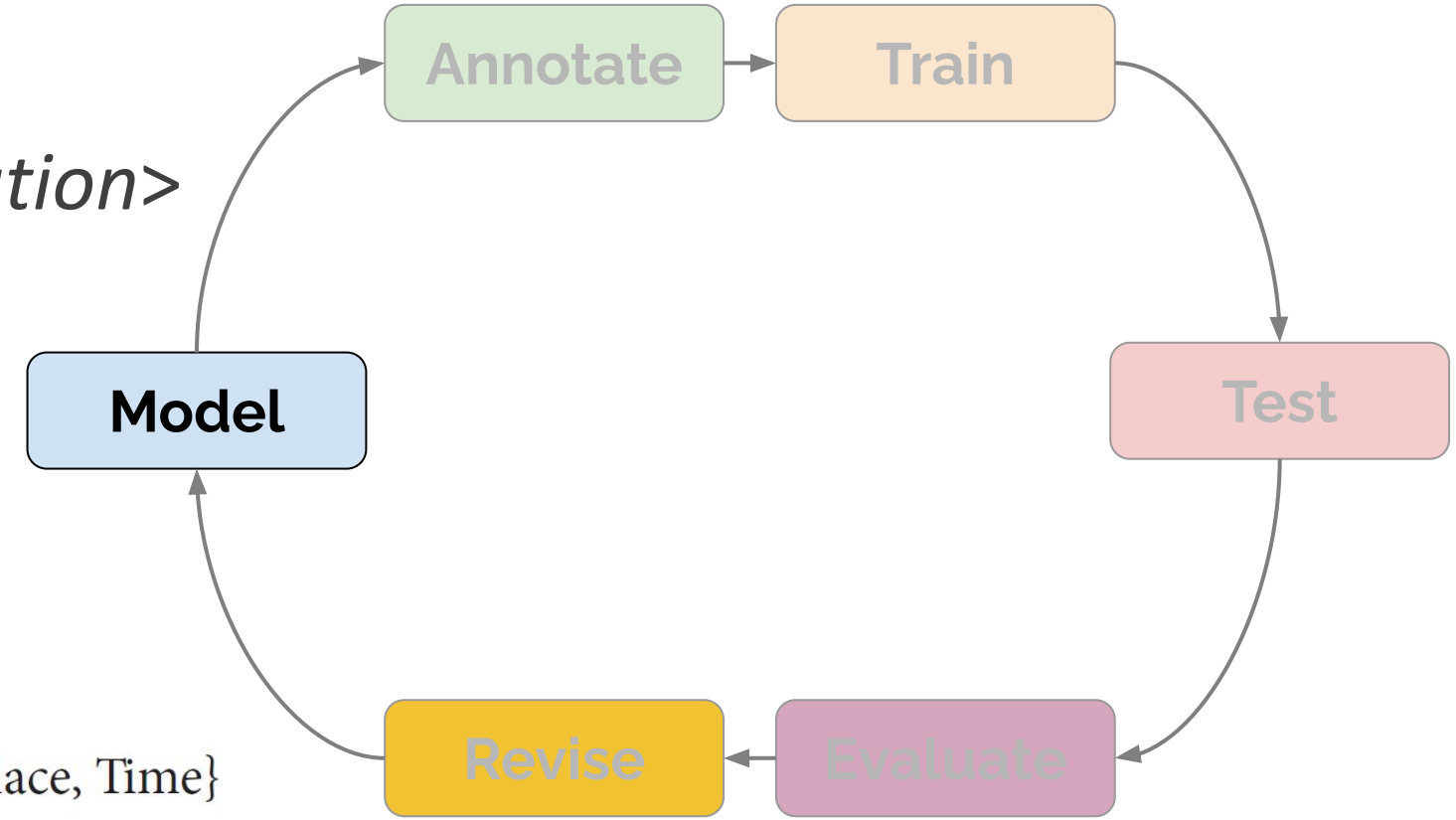
MATTER cycle



Model the phenomenon
Annotate with the specification
Train algorithms
Test them on unseen data
Evaluate the results
Revise the model and algorithms

MATTER cycle

A Model is made of:
*<**T**erms, **R**elations, **I**nterpretation>*



T = {Document_type, Spam, Not-Spam}

R = {Document_type ::= Spam | Not-Spam}

I = {Spam = "something we don't want!",
Not-Spam = "something we do want!"}

T = {Named_Entity, Organization, Person, Place, Time}

R = {Named_Entity ::= Organization | Person | Place | Time}

I = {Organization = "list of organizations in a database", Person = "list of people in a database", Place = "list of countries, geographic locations, etc.", Time = "all possible dates on the calendar"}

The Specification (spec)

- The model is captured by a specification, or spec. But what does a spec look like?
- You have the goals for your annotation project. Where do you start? How do you turn a goal into a model?
- What form should your model take? Are there standardized ways to structure the phenomena?
- How do you take someone else's standard and use it to create a specification?
- What do you do if there are no existing specifications, definitions, or standards for the kinds of phenomena you are trying to identify and model?
- How do you determine when a feature in your description is an element in the spec versus an attribute on an element?

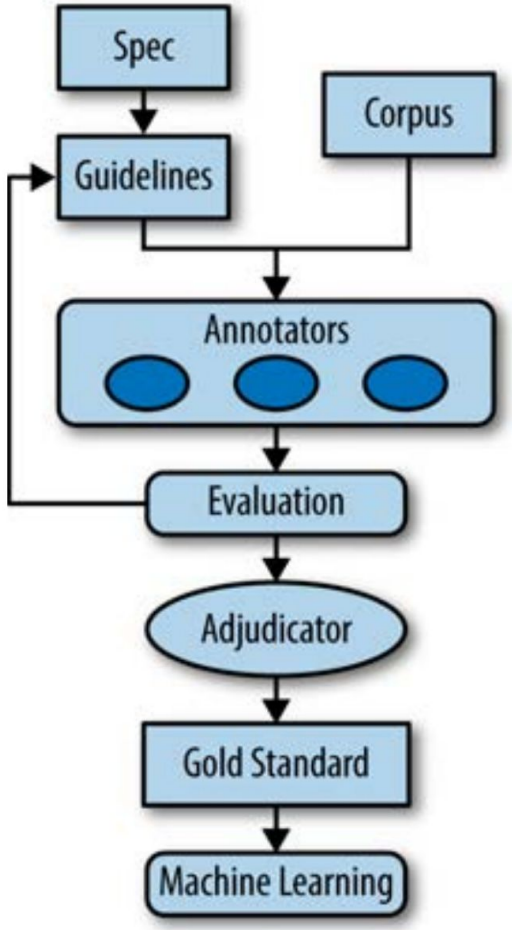
The spec is the concrete representation of your model. So, whereas the model is an abstract idea of what information you want your annotation to capture, and the interpretation of that information, the spec turns those abstract ideas into tags and attributes that will be applied to your corpus.

(Pustejovsky and Stubbs 2012, 67)

Annotation Specification

Why use a specification?

1. Scale
2. Generality



(Pustejovsky and Stubbs 2012, 106)

```
<?xml version="1.0"?>
<!DOCTYPE note SYSTEM "note.dtd">
<note>
 <to>Tove</to>
 <from>Jani</from>
 <heading>Reminder</heading>
 <body>Don't forget me this weekend!</body>
</note>
```

Model & Specification

Model = <***T****erms*, ***R****elations*, ***I****nterpretation*>

The annotation model is represented in XML.

XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use.

```
<?xml version="1.0" encoding="UTF-8"?>
<message>
   <warning>
      Hello World
   <!--missing </warning> -->
</message>
```

Specification

```
<!ELEMENT film_title ( #PCDATA ) >
<!ELEMENT director ( #PCDATA ) >
<!ELEMENT writer ( #PCDATA ) >
<!ELEMENT actor ( #PCDATA ) >
<!ELEMENT character ( #PCDATA ) >
```

DTD - A concrete information structure of XML files

Model & Specification: example

<genre label='not very funny comedy'>

<genre label='comedy'>

</genre>

Film genre classification

To believe again...
garcia-22 17 February 2006

Love Actually is movie that helps you see how life redefine's it self.

Common and extraordinary lives are mixed together with such good taste that you tend to believe that you were probably wrong the last time you were angry because of someone else doings.

The cast gave us outstanding performances by Hugh Grant, Liam Neesom and Emma Thompson and characters (the rock star and his manager are just persons to love!).

Special chapter for the charming Keira Knightley that just have away herself... and that is to say probably a person very similar to the general and popular idea of what angel's are.

```
<!ELEMENT genre ( #PCDATA ) >
<!ATTLIST genre label ( Action | Adventure | Animation | Biography | Comedy |
```

Model & Specification: example (II)

<genre label='comedy'>

To believe again...
garcia-22 17 February 2006

Love Actually is movie that helps you see how life redefine's it self.

Common and extraordinary lives are mixed together with such good taste that you tend to believe that you were probably wrong the last time you were angry because of someone else doings.

The cast gave us outstanding performances by Hugh Grant, Liam Neesom and Emma Thompson and characters (the rock star and his manager are just persons to love!).

Special chapter for the charming Keira Knightley that just have away herself... and that is to say probably a person very similar to the general and popular idea of what angel's are.

</genre>

Film genre classification
+ named entities

```
<!ELEMENT film_title ( #PCDATA ) >
<!ELEMENT director ( #PCDATA ) >
<!ELEMENT writer ( #PCDATA ) >
<!ELEMENT actor ( #PCDATA ) >
<!ELEMENT character ( #PCDATA ) >
```

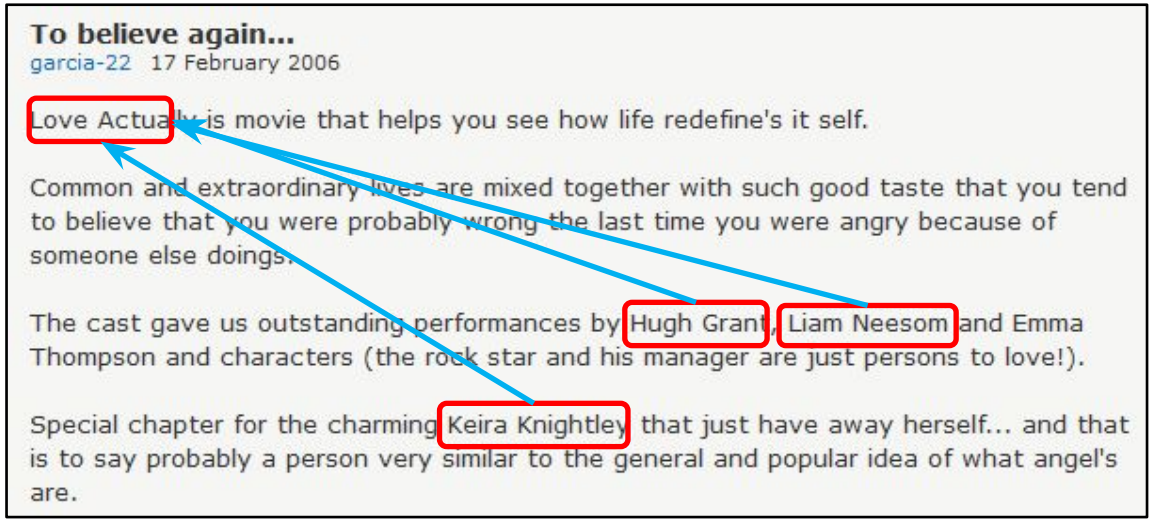
```
<actor>Hugh Grant</actor>
```

```
<named_entity role='actor'>Hugh Grant</named_entity>
```

Model & Specification: example (II)

<genre label='not very funny comedy'>

<genre label='comedy'>



</genre>

Film genre classification

+ named entities

+ semantic roles

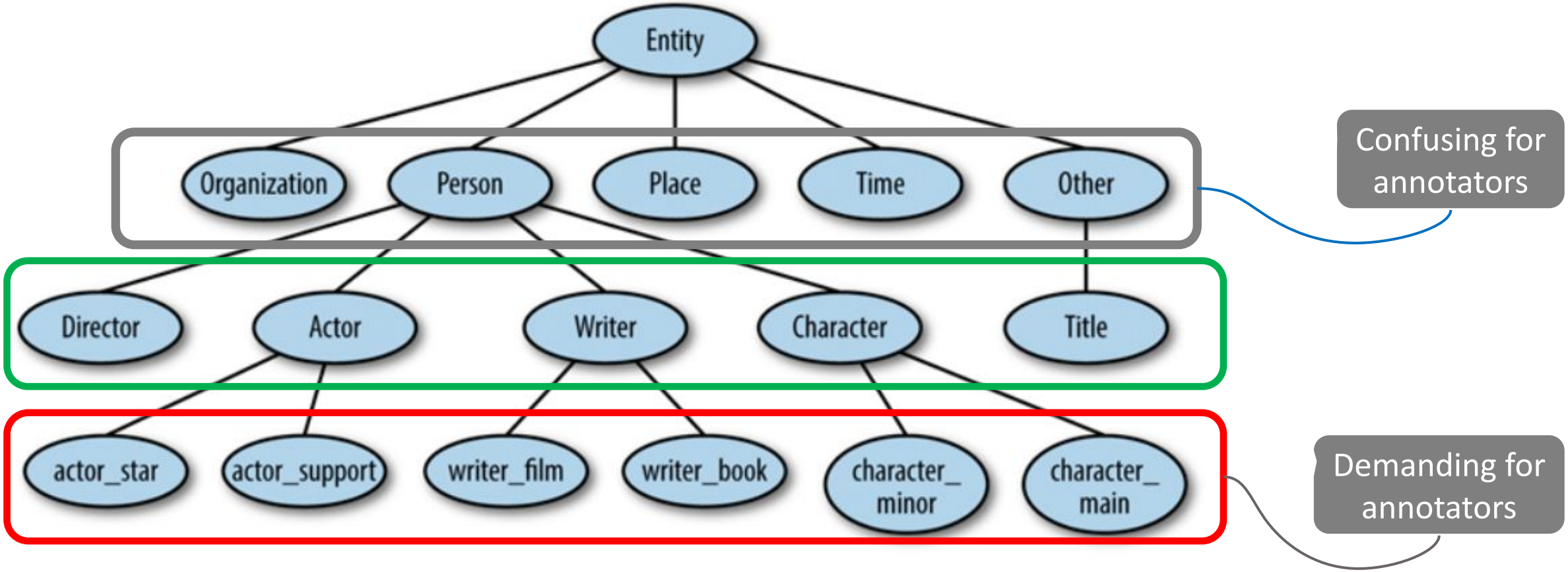
```
<!ELEMENT sem_role ( EMPTY ) >
<!ATTLIST sem_role from IDREF >
<!ATTLIST sem_role to IDREF >
<!ATTLIST sem_role label (acts_in |
  acts_as | directs | writes | chara
```

```
<acts_in from='ID1' to='ID2'/>
```

```
<sem_role from='ID1' to='ID2' label='acts_in'/>
```

Generality vs Specificity

Film genre classification
+ named entities



Using existing models & spec

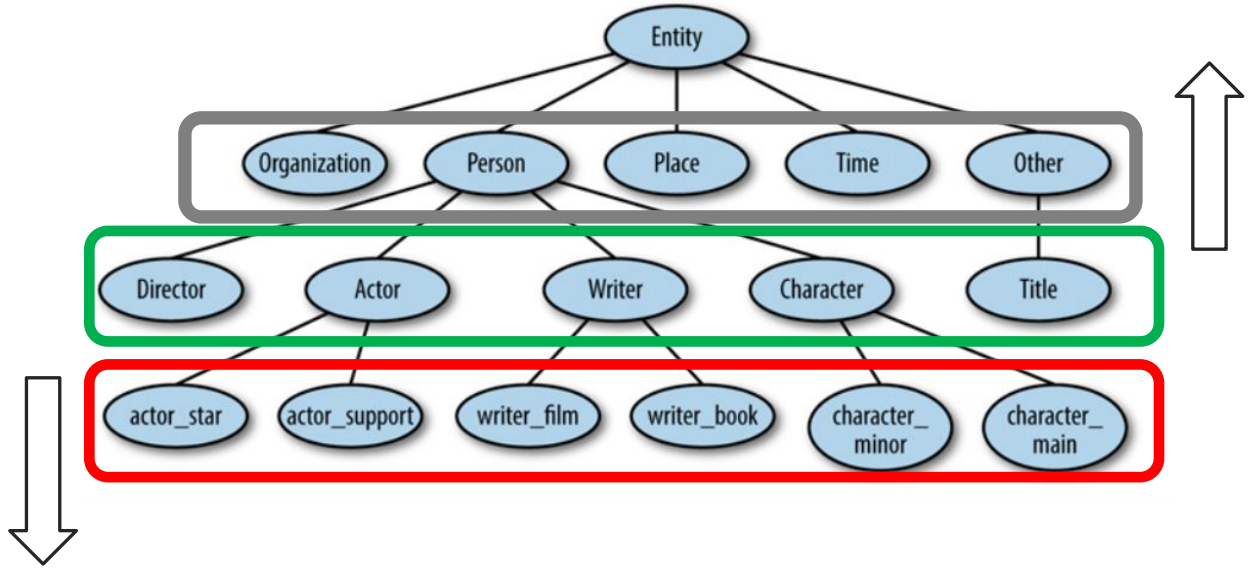
Existing annotation software

Existing data

Existing annotation guidelines

Adapting existing

- Make specific
- Make general
- Borrow ideas



Token Level Annotation

Text extent annotation

Sections of the text to be given distinct labels

- *POS tagging*
- *Word sense disambiguation*
- *Named entity*

A. Inline annotation

B. Stand-off: location in a sentence

C. Stand-off: character location

Mohandas Gandhi (1869 - 1948)

Otherwise known as Mahatma ('Great-Soul'), Gandhi was the leader of the Indian nationalist movement against British rule, and is widely considered the father of his country. During his political career, he won wide approval for his doctrine of non-violent protest to achieve political and social progress.

After university, Gandhi went to London to train as a barrister. There he met English socialists and Fabians such as George Bernard Shaw, whose ideas contributed greatly to the shaping of his personality and politics. He returned to India in 1891, then accepted

Standoff vs Inline Annotation

Standoff

```
The little cat drinks milk.

12,14,noun
```

Inline

```
The little <noun>cat</noun>
drinks milk.
```

Inline annotation

- Widely used (simplicity)
- Not always easy to read annotation
- Changes original text
- Impractical for combining >1 annotations

The Massachusetts State House in Boston, MA houses the offices of many important state figures, including Governor Deval Patrick and those of the Massachusetts General Court

<NE id="i0" type="building">**The Massachusetts State House**</NE> **in** <NE id="i1" type="city">**Boston, MA**</NE> **houses the offices of many important state figures, including** <NE id="i2" type="title">**Governor**</NE> <NE id="i3" type="person">**Deval Patrick**</NE> **and those of the** <NE id="i4" type="organization">**Massachusetts General Court**</NE>

Stand-off annotation: Tokens

- First: tokenize text
- Token positions as coordinates for location

TOKEN	TOKEN_ID
“	1
From	2
the	3
beginning	4
,	5
…	…
unit	31
.	32

TOKEN	SENT_ID	TOKEN_ID
“	1	1
From	1	2
the	1	3
beginning	1	4
,	1	5
…		
unit	1	31
.	1	32
Then	2	1
…		

Annotation example: POS tagging

POS_TAG	SENT_ID	TOKEN_ID
“	1	1
IN	1	2
DT	1	3
NN	1	4
…		

Stand-off annotation: Tokens

- First: tokenize text
- Token positions as coordinates for location
- Token spanning tags are allowed
- Not easy to recover original data
- Pairing annotation and original data is tough
- Not suitable for morpheme-level annotation

Annotation example: Named entities

TOKEN	SENT_ID	TOKEN_ID
The	1	1
Massachusetts	1	2
State	1	3
House	1	4
in	1	5
Boston	1	6
,	1	7
MA	1	8
houses	1	9
...		

TAG	START_SENT_ID	START_TOKEN_ID	END_SENT_ID	END_TOKEN_ID
NE_building	1	1	1	4
NE_city	1	6	1	8

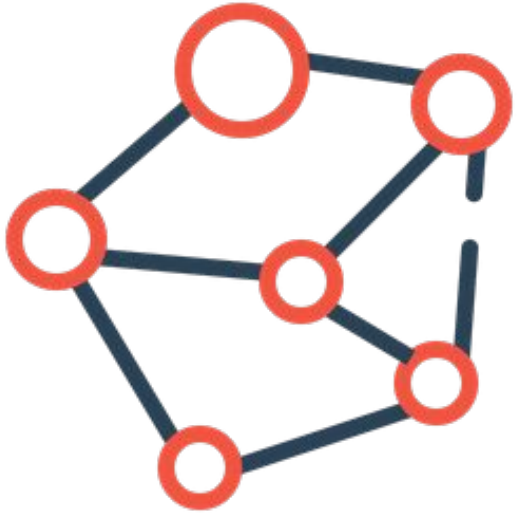
Stand-off annotation: Characters

- Read text character by character
- Character positions as coordinates for location
- Token spanning tags are allowed
- Original data is intact

The Massachusetts State House in Boston, MA houses the offices of many important state figures, including Governor Deval Patrick and those of the Massachusetts General Court

<NE id="N0" start="5" end="31" text="**Massachusetts State House**“ type="**building**" />
<NE id="N1" start="35" end="45" text="**Boston, MA**" type="**city**" />
<NE id="N2" start="109" end="117" text="**Governor**" type="**title**" />
<NE id="N3" start="118" end="131" text="**Deval Patrick**" type="**person**" />
<NE id="N4" start="150" end="177" text="**Massachusetts General Court**“ type="**organization**" />

Linked extent annotation



- Defined over IDs (opposed to extents in the text)
- Non-extent annotation *per se*

The Massachusetts State House in Boston, MA houses the offices of many important state figures, including Governor Deval Patrick and those of the Massachusetts General Court

```
<NE id="N0" start="5" end="31" text="Massachusetts State House“ type="building" />
<NE id="N1" start="35" end="45" text="Boston, MA" type="city" />
<NE id="N2" start="109" end="117" text="Governor" type="title" />
<NE id="N3" start="118" end="131" text="Deval Patrick" type="person" />
<NE id="N4" start="150" end="177" text="Massachusetts General Court“ type="organization" />
```

```
<L-LINK id="L0" fromID="N0" toID="N1" relationship="inside" />
<L-LINK id="L1" fromID="N4" toID="N0" relationship="inside" />
```

Semantic roles

Annotation Standards

Standards

ISO standards (committee-driven)

- Interoperability

Linguistic Annotation Framework

- XML-based *dump* format for interoperability
- Annotation is kept separately
- Separate file for each annotation level
- Hierarchical info also as flat structure
- Allow merging overlapping annotations
- Established labels for linguistic annotation
  - Data Category Registry

Community-driven

- Being first
- Has large community
- Existing datasets
- Part of the literature

An ISO standard

ISO standards are created with the intent of interoperability, which sets them apart from other de facto standards, as those often become the go-to representation simply because they were there first, or were used by a large community at the outset and gradually became ingrained in the literature. While this doesn't mean that non-ISO standards are inherently problematic, it does mean that they may not have been created with interoperability in mind.

(Pustejovsky and Stubbs, 81)

Linguistic Annotation Framework

How can a model be flexible enough to encompass all of the different types of annotation tasks? LAF takes a two-pronged approach to standardization. First, it focuses on the structure of the data, rather than the content. Specifically, the LAF standard allows for annotations to be represented in any format that the task organizers like, so long as it can be transmuted into LAF's XML-based "dump format," which acts as an interface for all manner of annotations. The dump format has the following qualities (Ide and Romary 2006):

(Pustejovsky and Stubbs, 81)

The other side of the approach that LAF takes toward standardization is encouraging researchers to use established labels for linguistic annotation. This means that instead of just creating your own set of POS or NE tags, you can go to the Data Category Registry (DCR) for definitions of existing tags, and use those to model your own annotation task. Alternatively, you can name your tag whatever you want, but when transmuting to the dump format, you would provide information about what tags in the DCR your own tags are equivalent to. This will help other people merge existing annotations, because

(Pustejovsky and Stubbs, 81)

LAF Focuses on the Structure

1. Annotations are separate from the text.
2. Each annotation level is separate.
3. Hierarchical annotations use XML.
4. When annotations are merged, over must be integrated.

Apply & adopt annotation standards

What you want your annotators to do

Decide ***how*** you want them to do it

Keep your **data easily accessible**

Annotation types:

- Metadata annotation, document labeling
- Extent tagging
- Tag linking

JSON vs XML

```
<DOCUMENT>
<NE id="N0" start="5" end="31" text="Massachusetts State House" type="building" />
<NE id="N1" start="35" end="45" text="Boston, MA" type="city" />
<NE id="N2" start="109" end="117" text="Governor" type="title" />
<NE id="N3" start="118" end="131" text="Deval Patrick" type="person" />
<NE id="N4" start="150" end="177" text="Massachusetts General Court" type="organization" />
</DOCUMENT>
```

```
{
   "text": "The Massachusetts State House in Boston, MA houses the offices of many important state figures, including Governor Deval Patrick
and those of the Massachusetts General Court",
   "entities": [
       {
           "id": "N0",
           "start": 5,
           "end": 31,
           "text": "Massachusetts State House",
           "type": "building"
       },
       {
           "id": "N1",
           "start": 35,
           "end": 45,
           "text": "Boston, MA",
           "type": "city"
       },
       {
           "id": "N2",
           "start": 109,
           "end": 117,
           "text": "Governor",
           "type": "title"
       },
       {
           "id": "N3",
           "start": 118,
           "end": 131,
```

XML & JSON & JSON Lines

Why XML
- Limitless Customisation
- Inline annotations are more interpretable
- Typeless
- Supports namespaces
- More secure

Why JSON
- Easy (familiar with minimal dependencies)
- Fast
- Free
- No mapping

Why JSON lines
- Appendable
- Adaptable

```
# JSON

[
{"text": "Annotation is always fun.", "labels": [[0, 10]]},
{"text": "Annotation is always fun.", "labels": [[0, 10]]},
{"text": "Annotation is always fun.", "labels": [[0, 10]]},
{"text": "Annotation is always fun.", "labels": [[0, 10]]}
]

# JSON LINES

{"text": "Annotation is always fun.", "labels": [[0, 10]]}
{"text": "Annotation is always fun.", "labels": [[0, 10]]}
{"text": "Annotation is always fun.", "labels": [[0, 10]]}
{"text": "Annotation is always fun.", "labels": [[0, 10]]}
```

Document Level Annotation

What is a document?

Document-level annotation

Classifying movie reviews:

- *Positive* 
- *Neural* 
- *Negative* 

	A	B
1	review001	positive
2	review002	neutral
3	review003	negative
4	review004	positive
5	review005	negative
6	review006	
7	review007	
8	review008	

A. Plain text file: Filename, label

~~B. SQL table per label with filename entries~~

C. Directory per label with files inside it

D. Add labels to the filenames

~~E. Add labels inside the files~~

Document-level annotation

Assign genres to movies:

- *Action*
- *…*
- *Thriller*

```
{
    "title": "The Last of the Mohicans",
    "year": "1992",
    "cast": [
        "Michael Douglas",
        "Catherine O'Hara",
        "John Heard"
    ],
    "genres": [
        "Action",
        "Adventure",
        "Comedy"
    ]
}
```

Take a break?

Parallel Meaning Bank (PMB)

Get a large collection of documents

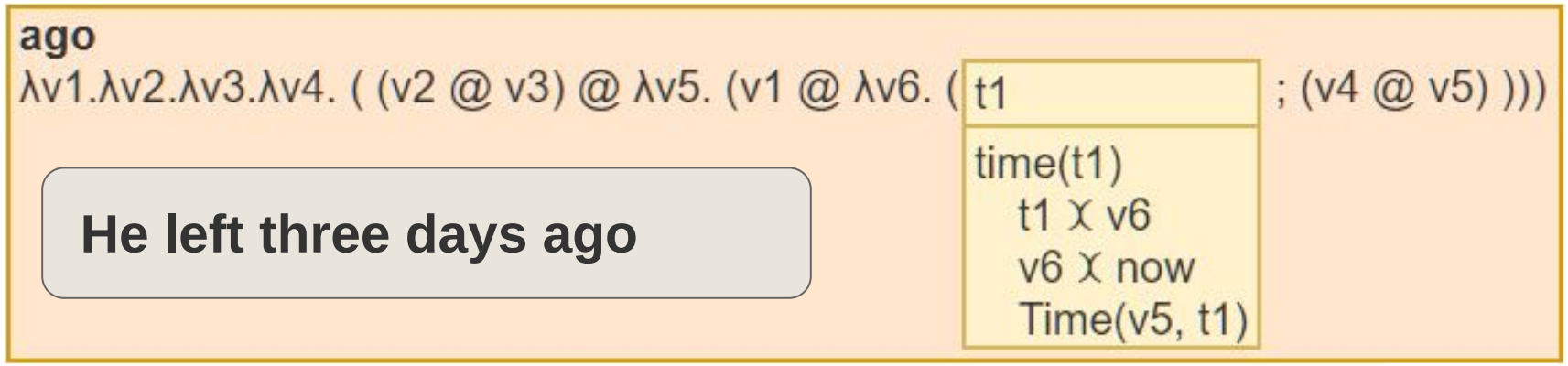
- Study formal semantics of wide-coverage texts
- Automatically learn Meaning Representations (MRs)

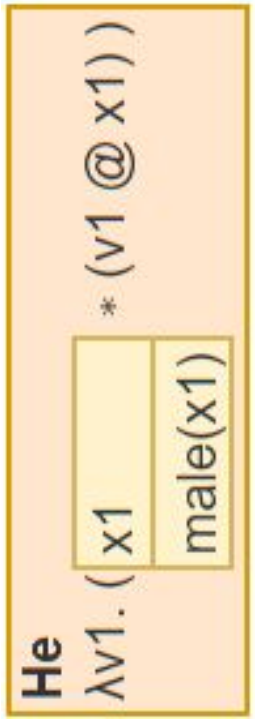
Parallel data

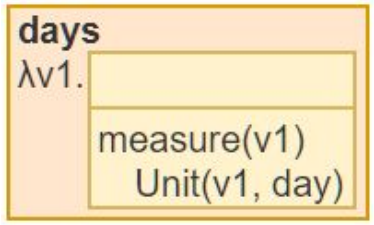
- Bridging done via compositionality & alignments
- Alleviate compositional semantics by rich annotations

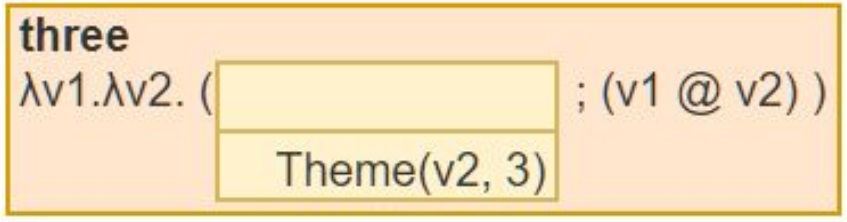
Compositional semantics

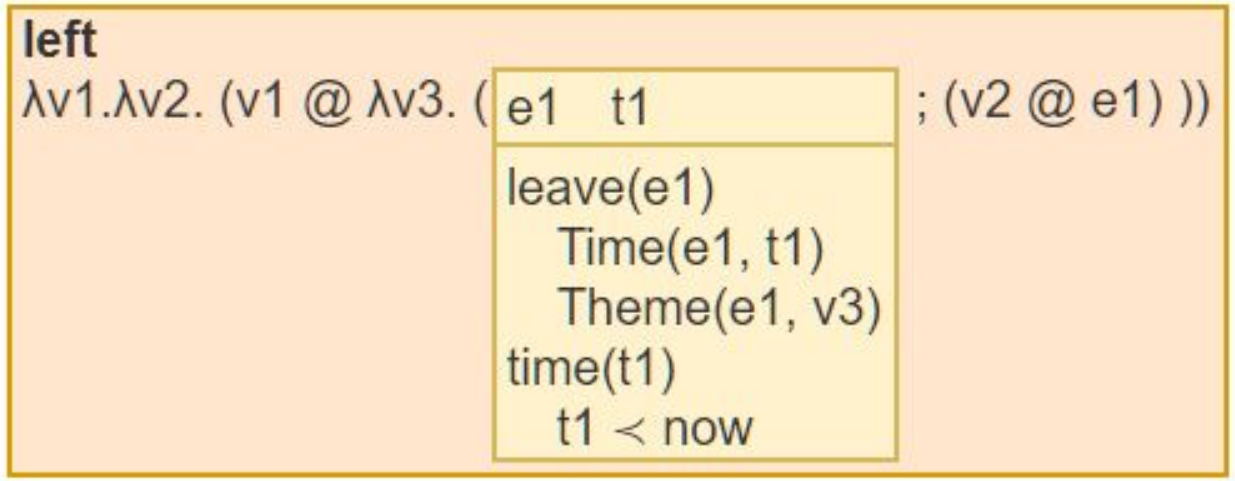
Lexical semantic = building blocks









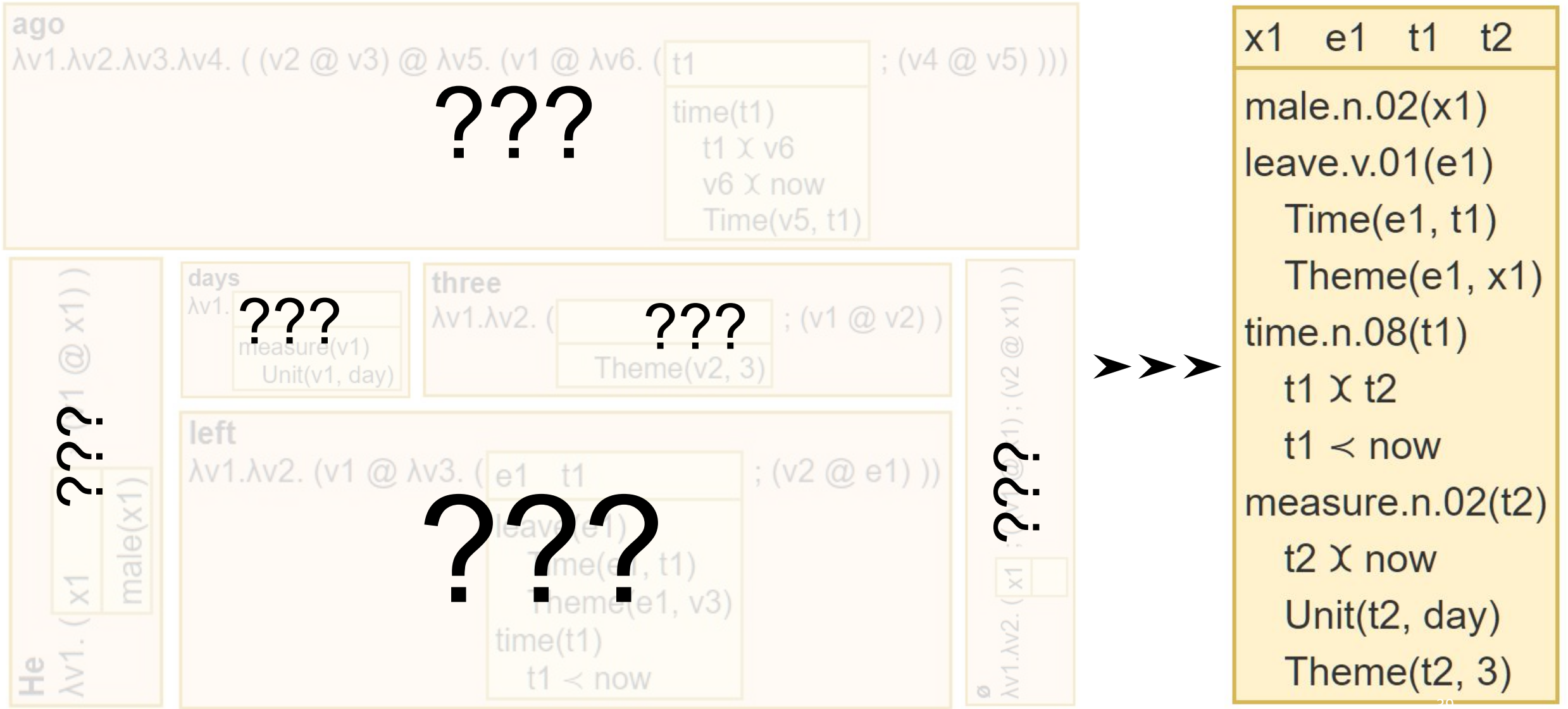


➤➤➤

x1	e1	t1	t2
male.n.02(x1)			
leave.v.01(e1)			
Time(e1, t1)			
Theme(e1, x1)			
time.n.08(t1)			
t1 ⊀ t2			
t1 ≺ now			
measure.n.02(t2)			
t2 ⊀ now			
Unit(t2, day)			
Theme(t2, 3)			

Compositional semantics

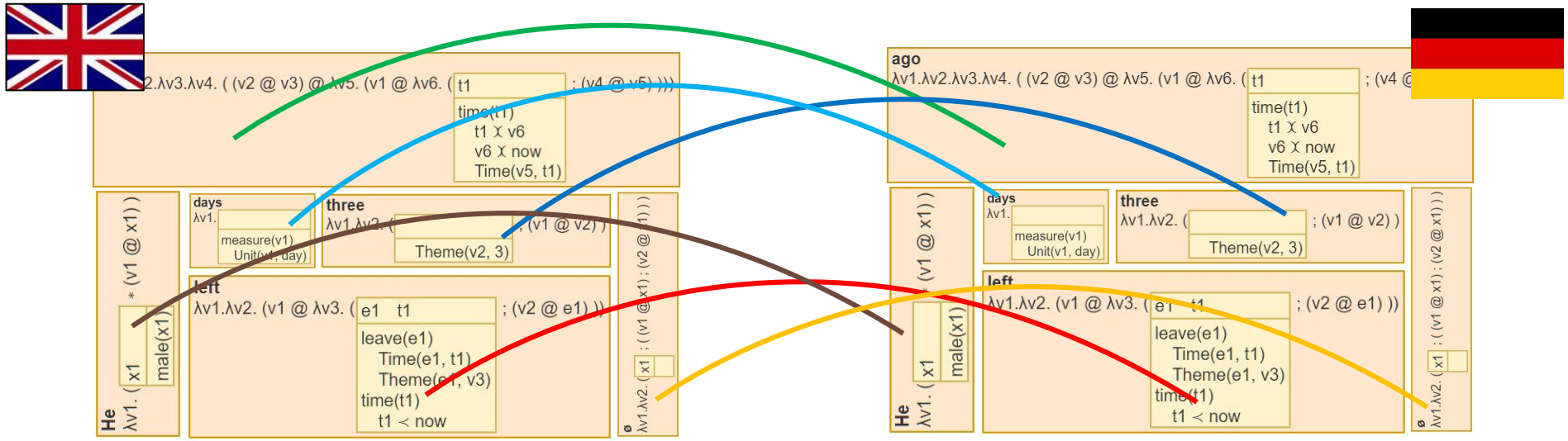
Lexical semantic = building blocks



Compositionality + alignments

He came back at 5 o'clock.

Er kam um fünf Uhr zurück.



compositionality

x1	e1	t1	t2
male.n.02(x1)			
leave.v.01(e1)			
Time(e1, t1)			
Theme(e1, x1)			
time.n.08(t1)			
t1 ϰ t2			
t1 ≺ now			
measure.n.02(t2)			
t2 ϰ now			
Unit(t2, day)			
Theme(t2, 3)			

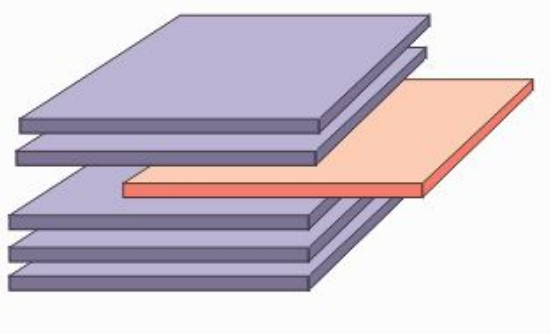
~

x1	e1	t1	t2
male.n.02(x1)			
leave.v.01(e1)			
Time(e1, t1)			
Theme(e1, x1)			
time.n.08(t1)			
t1 ϰ t2			
t1 ≺ now			
measure.n.02(t2)			
t2 ϰ now			
Unit(t2, day)			
Theme(t2, 3)			

compositionality

Use Cases: Annotation Specification

Rich token-based annotation layers



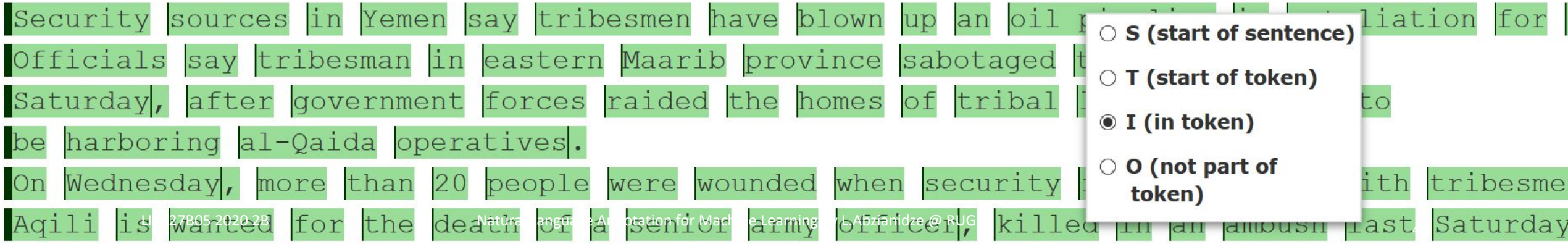
- Segmentation
- Semantic tagging (~~part-of-speech tagging~~)
- Symbolization (~~lemmatization~~)
- Word sense disambiguation (Wordnet 3.0; Miller, 1994)
- Semantic role labeling (Verbnet roles; Bonial et al, 2011)
- Syntactic parsing (Combinatory Categorial Grammar)
- Semantic parsing (Discourse Representation Theory)

Segmentation (sentence & token)

- Split texts into sentences
  - John said “I won’t go. I am lazy”.
- Split sentences into “meaningful atoms/words”
  - San~Diego, Secretary~of~State, Royal~Bank~of~Scotland, …
  - Baseball~club, knitting~needles, …
  - un|happy, im|possible, dis|agree, …
  - ten-year-old, data-driven, New~York-based …

Segmentation (IOB method)

- Character-based, i.e. label characters
- Each characters gets one of the four labels:
  - **S** – start of a sentence
  - **T** – start of a token
  - **I** – inside a token
  - **O** – outside of a token



Segmentation (IOB method)

```
His cell phone is off.
```

```
His cell␣phone is off
```

- **S** – start of a sentence
- **T** – start of a token
- **I** – inside a token
- **O** – outside of a token

```
0 3 1001 His
4 14 1002 cell phone
15 17 1003 is
18 21 1004 off
21 22 1005 .
out/p05/d2458/en.tok.off (END)
```

```
72 S
105 I
115 I
32 O
99 T
101 I
108 I
108 I
32 I
112 I
104 I
111 I
110 I
101 I
32 O
105 T
115 I
32 O
111 T
102 I
102 I
46 T
13 O
10 O
out/p05/d2458/en.tok.iob (END)
```

Symbolization

Mapping tokens to non-logical symbols

- Lemmatization:
  morphological analysis
- Normalization:
  mapping to a canonical form

```
male
cellphone
be
off
.
out/p05/d2458/en.lemma (END)
```

```
0 3 1001 His
4 14 1002 cell phone
15 17 1003 is
18 21 1004 off
21 22 1005 .
out/p05/d2458/en.tok.off (END)
```

token	symbol
third	3
men	man
played	play
2:30 pm	14:30
2,5 million	2500000
km	kilometer

Word sense disambiguation

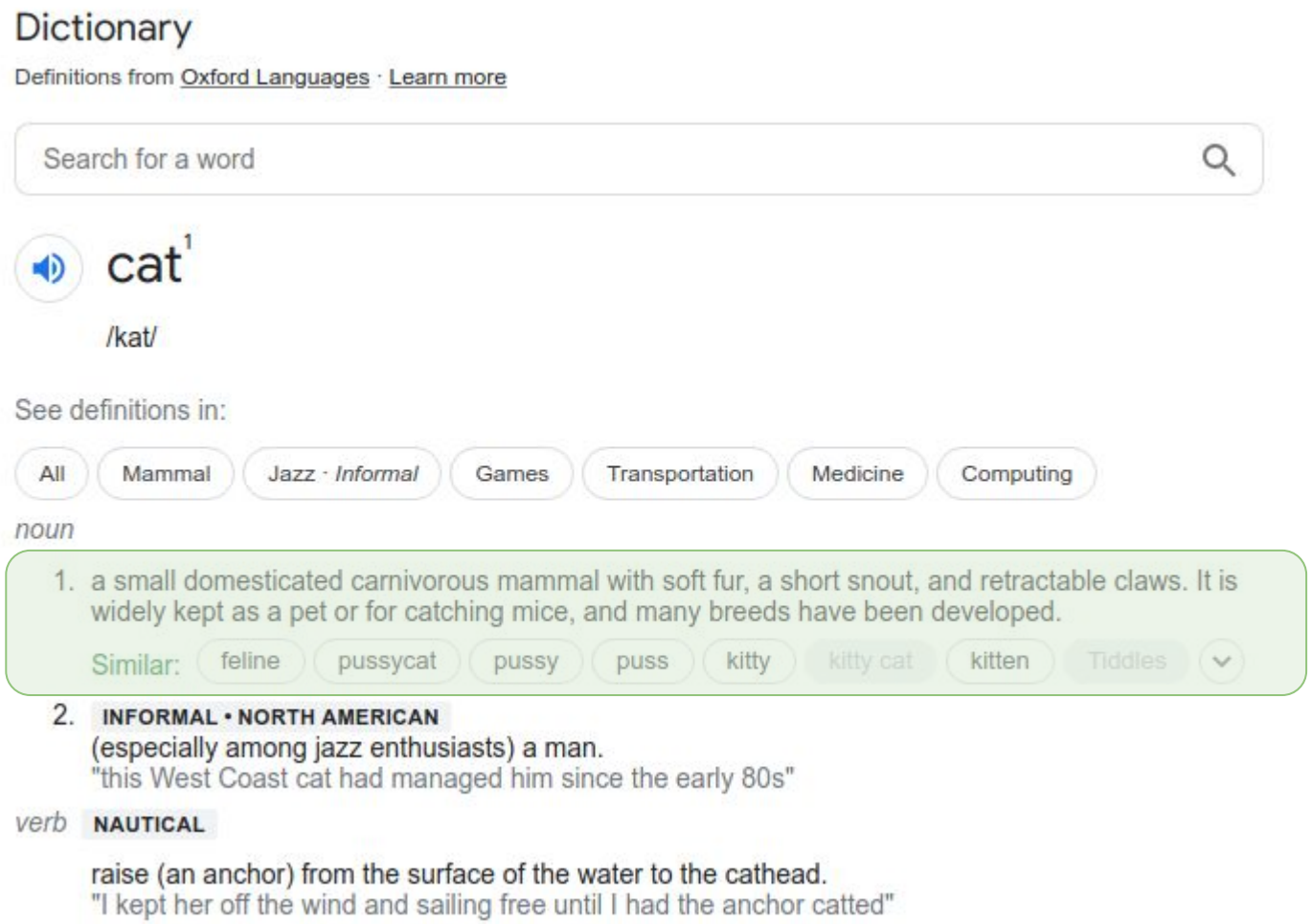
Assigning sense numbers to non-logical symbols

Not all of them get a sense number

- Noun concepts
  Named entities
  Pronouns (gender)
- Verb concepts
- Adjective concepts
- Adverb concepts

token	symbol	sense
third	3	O
men	man	man.n.02
played	play	play.v.02
2:30 pm	14:30	O
Kraft	kraft	company.n.01
she	female	female.n.02

A word sense



Dictionary

Definitions from Oxford Languages · Learn more

Search for a word

cat[1]

/kat/

See definitions in:

All | Mammal | Jazz · *Informal* | Games | Transportation | Medicine | Computing

noun

1. a small domesticated carnivorous mammal with soft fur, a short snout, and retractable claws. It is widely kept as a pet or for catching mice, and many breeds have been developed.

   Similar: feline | pussycat | pussy | puss | kitty | kitty cat | kitten | Tiddles

2. **INFORMAL • NORTH AMERICAN**
   (especially among jazz enthusiasts) a man.
   "this West Coast cat had managed him since the early 80s"

verb **NAUTICAL**

raise (an anchor) from the surface of the water to the cathead.
"I kept her off the wind and sailing free until I had the anchor catted"

Syntactic parsing

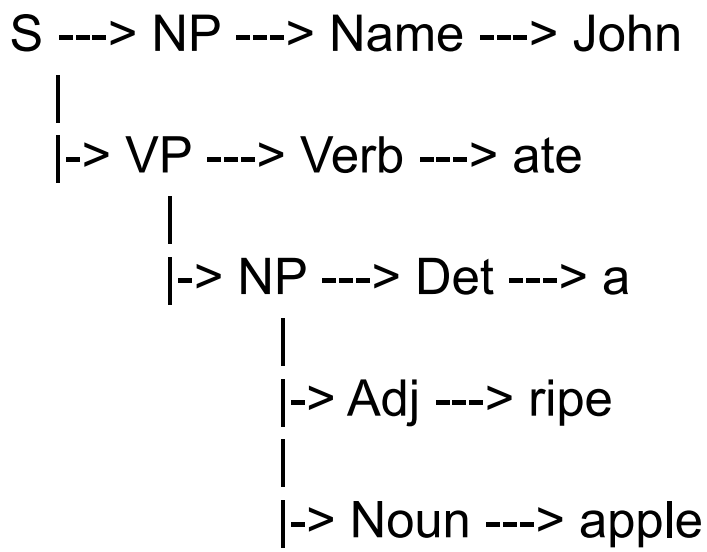
with Combinatory categorial grammar (CCG; Steedman, 2000)

- Goes well with compositional semantics
- Lexicalized grammar
- Efficient and wide-coverage parsers are available
  - C&C (Clark & Curran, 2007)
  - EasyCCG (Lewis & Steedmam, 2014)
  - DepCCG (Yoshikawa et al, 2017)
  - EasySRL (Lewis et al., 2016)

Compositional Semantics

"John ate a ripe apple."

Syntax tree :



Function Words vs Content Words

Function Words

articles

the and a.

pronouns

they, he, him, she, and her.

adpositions

in, under, towards, before, of, for, etc.

conjunctions

and and but

…

Content Words

nouns

things, objects, people

adjectives

characteristics, qualities

…

Syntactic parsing (II)

His	cell~phone	is	off	.
NP/(N/PP)	N/PP	(S[dcl]\NP)/(S[adj]\NP)	S[adj]\NP	S[dcl]\S[dcl]

>

>

His cell~phone — NP

is off — S[dcl]\NP

<

His cell~phone is off — S[dcl]

<

His cell~phone is off . — S[dcl]

```
0 3 1001 His
4 14 1002 cell phone
15 17 1003 is
18 21 1004 off
21 22 1005 .
out/p05/d2458/en.tok.off (END)
```

```
NP/(N/PP)
N/PP
(S[dcl]\NP)/(S[adj]\NP)
S[adj]\NP
.
out/p05/d2458/en.cats (E
```

```
ccg(1,
 ba(s:dcl,
  fa(np,
   t(np/(n/pp), 'His', [lemma:'his']),
   t(n/pp, 'cell~phone', [lemma:'cell~phone'])),
  rp(s:dcl\np,
   fa(s:dcl\np,
    t((s:dcl\np)/(s:adj\np), 'is', [lemma:'is']),
    t(s:adj\np, 'off', [lemma:'off'])),
   t(., '.', [lemma:'.'])))).

out/p05/d2458/en.parse (END)
```

Semantic roles

His	cell~phone	is	off	.
[User]	[]	[]	[Attribute]	[]
NP/(N/PP)	N/PP	(S[dcl]\NP)/(S[adj]\NP)	S[adj]\NP	S[dcl]\S[dcl]

- The roles are mainly borrowed from VerbNet
- Only tokens with function categories can have roles

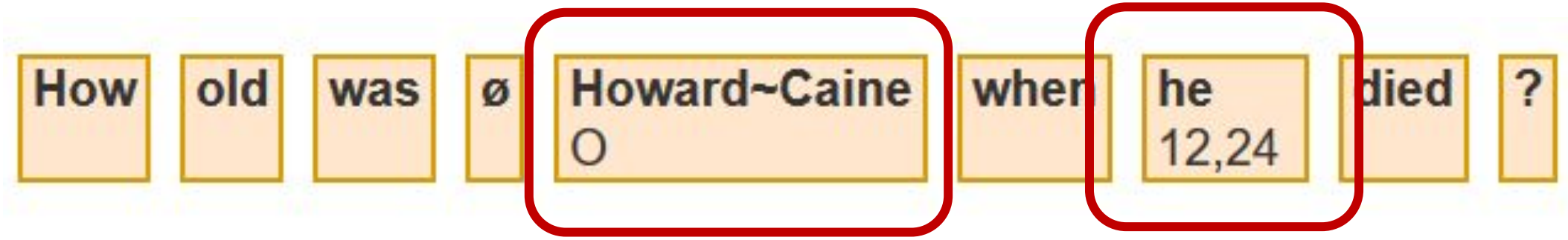
```
0 3 1001 His
4 14 1002 cell phone
15 17 1003 is
18 21 1004 off
21 22 1005 .
out/p05/d2458/en.tok.off (END)
```

```
[User]


[Attribute]

out/p05/d2458/en.roles (END)
```

References




```
0 3 1001 How
4 7 1002 old
8 11 1003 was
12 24 1004 Howard Caine
25 29 1005 when
30 32 1006 he
33 37 1007 died
37 38 1008 ?
out/p71/d1390/en.tok.off (END)
```

```
12,24
out/p71/d1390/en.antecedent (END)
```

Model & Specification

- Generality vs specificity
- Interoperability & standardization

Kinds of annotations

- Document-level labels
- Extent tags: inline, stand-off tokens, stand-off charcters
- Link tags

Annotations in the Parallel Meaning Bank