Syntactic Analysis

Daniel Zeman

January 3, 2024
Outline

1. Constituents vs. Dependencies

2. Universal Dependencies
   - A Tour through UD Syntax
   - Nonverbal Predicate and Copula
   - Core Arguments vs. Oblique Dependents
   - Ellipsis and Enhanced UD
Constituents vs. Dependencies
Syntactic Structure

- Different shapes in different theories
- Typically a tree
  - Constituents (phrase tree structure)
  - Dependencies (dependency tree structure)
(S (NP (N Paul)) (VP (V gave) (NP (N Peter)) (NP (C two) (N pears))))

Constituent Tree

S
  NP
    N
      Paul
    VP
      V
        gave
      NP
        N
          Peter
        C
          two
        NP
          N
            pears
Paul gave Peter two pears.
[#,0] ([gave,2] ([Paul,1], [Peter,3], [pears,5] ([two,4])), [,6])

Dependency Tree with Labels

# / AuxS

- gave / Pred

  - Paul / Sb
  - Peter / Obj
  - pears / Obj

  - two / Atr
Paul gave Peter two pears
Constituents vs. Dependencies

• The two models are interconnected

• Sentence divided to phrases (constituents)
  • Recursive: phrases divided to smaller phrases
  • The smallest phrases are words

• There are dependencies (relations) between words (constituents)
  • Head of phrase = governing node, parent node
  • The other nodes are dependent nodes, children of the head
Phrase vs. Dependency Trees

Constituents vs. Dependencies

Universal Dependencies
Phrase vs. Dependency Trees

• Phrase trees
  • Usually do not mark the **head**
  • May not mark the **function** of the constituent in the superordinate constituent
Phrase vs. Dependency Trees

• Phrase trees
  • Usually do not mark the head
  • May not mark the function of the constituent in the superordinate constituent

• Dependency trees
  • Do not show nonterminals (phrase types)
    • Nor any other phrase-level features
  • Do not show “how the sentence is generated” (order, recursion, proximity of constituents)
Example

Constituents vs. Dependencies

Universal Dependencies
Discontinuous Phrases

- Classical context-free grammar cannot describe them!
- They cannot be represented by bracketing.
- English example: *I found the best example ever.*
- Czech example: *(Soubor (se nepodařilo) otevřít).* “File couldn’t be opened.”

```
VP(nepodařilo)
  /      \       \       
VR(nepodařilo)   VP_{inf}(otevřít) \
  /       \   / \       \ 
T V  V_{inf} N  \\
se nepodařilo  otevřít soubor
```
• Dependency tree including word order (x-coordinate of nodes).
• Projection to the base: the vertical from the node crosses a dependency (nonprojective edge).
• Formally:
  • Dependency $([g, x_g], [d, x_d])$ where $x_w$ is the order of the word $w$ in the sentence.
  • There exists a node $[n, x_n]$ that $x_g < x_n < x_d$ or $x_d < x_n < x_g$ and $[n, x_n]$ is not in subtree rooted by $[g, x_g]$.
• Informally: The string spanned by the subtree of the governing node is discontinuous, it contains gaps.
Nonprojectivity Can Be Handled by a Dependency Tree!

```
nepodařilo / Pred
  se / AuxT  otevřít / Sb
    soubor / Obj
```
Nonprojectivity Can Be Handled by a Dependency Tree!
Universal Dependencies
Outline

1. Constituents vs. Dependencies

2. Universal Dependencies
   - A Tour through UD Syntax
   - Nonverbal Predicate and Copula
   - Core Arguments vs. Oblique Dependents
   - Ellipsis and Enhanced UD
Constituents vs. Dependencies
Constituents vs. Dependencies
My daughter bought some bread and cheese.
Universal Dependencies

http://universaldependencies.org/

Milestones:

- 2008-05 Interset (morphological features)
- 2012-05 Google Universal POS tags
- 2012-05 HamleDT (harmonized Prague-style treebanks)
- 2013-08 Google Universal Dependency Treebank
- 2014-05 Universal Stanford Dependencies
- 2014-04 EACL Göteborg, kick-off meeting of UD
- 2014-10 UD guidelines version 1
- 2015-01 released first 10 treebanks
- every ~6 months new release
- 2016-12 UD guidelines version 2
- 2017-05 CoNLL Shared Task in parsing UD
- 2018-06 second Shared Task
- every ~6 months new release
• Same things annotated same way across languages...
• ... while highlighting different coding strategies
Manning’s Law

The secret to understanding UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.

2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.

3. UD must be suitable for rapid, consistent annotation by a human annotator.

4. UD must be easily comprehended and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing. … it leads us to favor traditional grammar notions and terminology.

5. UD must be suitable for computer parsing with high accuracy.

6. UD must support well downstream language understanding tasks (relation extraction, reading comprehension, machine translation, …)

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
Manning’s Law

The secret to understanding UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.
2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
Manning’s Law

The secret to understanding UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.
2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. UD must be suitable for rapid, consistent annotation by a human annotator.

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
Manning’s Law

The secret to understanding UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.
2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. UD must be suitable for rapid, consistent annotation by a human annotator.
4. UD must be easily comprehended and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing. ... it leads us to favor traditional grammar notions and terminology.
5. UD must be suitable for computer parsing with high accuracy.
6. UD must support well downstream language understanding tasks (relation extraction, reading comprehension, machine translation, ...)

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
Manning’s Law

The secret to understanding UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.
2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. UD must be suitable for rapid, consistent annotation by a human annotator.
4. UD must be easily comprehended and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing. ... it leads us to favor traditional grammar notions and terminology.
5. UD must be suitable for computer parsing with high accuracy.

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
Manning’s Law

The secret to understanding UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.
2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. UD must be suitable for rapid, consistent annotation by a human annotator.
4. UD must be easily comprehended and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing. ... it leads us to favor traditional grammar notions and terminology.
5. UD must be suitable for computer parsing with high accuracy.
6. UD must support well downstream language understanding tasks (relation extraction, reading comprehension, machine translation, ...)

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
George killed the dragon

Constituents vs. Dependencies
George killed the dragon

Mharaigh Seoirse an dragan
Same Thing Same Way

- George killed the dragon
- Mharaigh Seoirse an dragan
- Jorge mató al dragón

Constituents vs. Dependencies
George killed the dragon

Mharaigh Seoirse an dragan

Jorge mató a el dragón
George killed the dragon

Mharaigh Seoirse an dragan

Jorge mató a el dragón

Draka zabil Jiří
Same Meaning ≠ Same Construction!

He killed the dragon.

**Constituents vs. Dependencies**
Same Meaning $\neq$ Same Construction!

He killed the dragon

The dragon was killed by him

Constituents vs. Dependencies
Same Meaning ≠ Same Construction!

He killed the dragon

The dragon was killed by him

His killing of the dragon

Constituents vs. Dependencies
Same Meaning ≠ Same Construction!

Constituents vs. Dependencies
राजा विष्णुशर्मान्
king Vishnusharma

आहूया प्रोवाच
having-summoned said

VerbForm=Conv VerbForm=Fin
राजा 
rājā
king

विष्णुशर्माणम्
viṣṇuśarmāṇam
Vishnusharma

आहूय
āhūya
having-summoned

provāca
said

VerbForm=Conv

VerbForm=Fin

VerbForm=Fin
Outline

1. Constituents vs. Dependencies

2. Universal Dependencies
   - A Tour through UD Syntax
   - Nonverbal Predicate and Copula
   - Core Arguments vs. Oblique Dependents
   - Ellipsis and Enhanced UD
The cat could have chased all the dogs down the street.

DET NOUN AUX AUX VERB DET DET NOUN ADP DET NOUN PUNCT
Syntax

The cat could have chased all the dogs down the street.

- Content words are related by dependency relations

- Function words attach to closest content words
- Punctuation attach to head of phrase or clause
- Not "dependency" in the strictly syntactic sense!
The cat could have chased all the dogs down the street.

• Content words are related by dependency relations
• Function words attach to closest content words
The cat could have chased all the dogs down the street.

• Content words are related by dependency relations
• Function words attach to closest content words
• Punctuation attach to head of phrase or clause
The cat could have chased all the dogs down the street.

Not "dependency" in the strictly syntactic sense!
The dog was chased by the cat.

Кучето се преследваше от котката.
The dog was chased by the cat.
The dog was chased by the cat.

Кучето беше преследвано от котката.
The dog was chased by the cat.
### Dependents of Clauses (Verbal or Not)

<table>
<thead>
<tr>
<th>Core</th>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsubj</td>
<td>csubj</td>
<td>advmod</td>
<td>aux</td>
<td></td>
</tr>
<tr>
<td>expl</td>
<td>advcl</td>
<td>discourse</td>
<td>cop</td>
<td></td>
</tr>
<tr>
<td>dislocated</td>
<td>advcl</td>
<td>mark</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vocative</td>
<td>obli</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Dependents of Verbs, Adjectives and Adverbs

<table>
<thead>
<tr>
<th>Core</th>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>ccomp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iobj</td>
<td>xcomp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-Core</td>
<td>obli</td>
<td>advcl</td>
<td>advmod</td>
</tr>
<tr>
<td>expl</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Dependents of Nominals

<table>
<thead>
<tr>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nmod</td>
<td>acl</td>
<td>amod</td>
<td>det</td>
</tr>
<tr>
<td>appos</td>
<td></td>
<td>nummod</td>
<td>case</td>
</tr>
</tbody>
</table>
### Dependents of Nominals

<table>
<thead>
<tr>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nmod</td>
<td>acl</td>
<td>amod</td>
<td>det</td>
</tr>
<tr>
<td>appos</td>
<td></td>
<td>nummod</td>
<td>case</td>
</tr>
<tr>
<td>compound</td>
<td></td>
<td></td>
<td>clf</td>
</tr>
<tr>
<td>flat</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Diagram:**

```
the American singer Johnny Cash , an icon of country music
DET ADJ NOUN PROPN PROPN PUNCT DET NOUN ADP NOUN NOUN
```

Constituents vs. Dependencies
Coordination

- Coordinate structures are headed by the first conjunct
  - Subsequent conjuncts depend on it via the `conj` relation
  - Conjunctions depend on the next conjunct via the `cc` relation
  - Punctuation marks depend on the next conjunct via the `punct` relation
But Some Languages Might Prefer the Opposite

Coordinate structures would be headed by the last conjunct
- Preceding conjuncts would depend on it via the conj relation
- Conjunctions would depend on the preceding conjunct
- Punctuation marks would depend on the preceding conjunct

Yıldırım, Erdoğan ve Akar
PROPN PUNCT PROPN CCONJ PROPN
### Multiword Expressions

<table>
<thead>
<tr>
<th>Relation</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>fixed</td>
<td>in spite of, as well as, ad hoc</td>
</tr>
<tr>
<td>flat</td>
<td>president Havel, New York, four thousand</td>
</tr>
<tr>
<td>compound</td>
<td>phone book, dress up</td>
</tr>
<tr>
<td>goeswith</td>
<td>notwith standing, with out</td>
</tr>
</tbody>
</table>

- UD annotation **almost** does not permit “words with spaces”
  - Multiword expressions are analyzed using special relations
  - The **fixed**, **flat** and **goeswith** relations are always head-initial
  - The **compound** relation reflects the internal structure
- Words with spaces may be allowed in v2:
  - Vietnamese (spaces delimit syllables, not words)
  - Numbers (“1 000 000”)
  - Possibly other approved cases, e.g. multi-word abbreviations
<table>
<thead>
<tr>
<th>Relation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>parataxis</td>
<td>Loosely linked clauses of same rank</td>
</tr>
<tr>
<td>list</td>
<td>Lists without syntactic structure</td>
</tr>
<tr>
<td>orphan</td>
<td>Orphans in ellipsis linked together</td>
</tr>
<tr>
<td>reparandum</td>
<td>Disfluency linked to (speech) repair</td>
</tr>
<tr>
<td>foreign</td>
<td>Elements within opaque stretches of code switching</td>
</tr>
<tr>
<td>dep</td>
<td>Unspecified dependency</td>
</tr>
<tr>
<td>root</td>
<td>Syntactically independent element of clause/phrase</td>
</tr>
</tbody>
</table>
• Language-specific relations are **subtypes** of universal relations added to capture important phenomena

• Subtyping permits us to “back off” to universal relations

### Language-specific Relation Subtypes

<table>
<thead>
<tr>
<th>Relation</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>acl:relcl</td>
<td>Relative clause</td>
</tr>
<tr>
<td>compound:prt</td>
<td>Verb particle (dress up)</td>
</tr>
<tr>
<td>nmod:poss</td>
<td>Possessive nominal (Mary’s book)</td>
</tr>
<tr>
<td>obl:agent</td>
<td>Agent in passive (saved by the bell)</td>
</tr>
<tr>
<td>cc:preconj</td>
<td>Preconjunction (both ... and)</td>
</tr>
<tr>
<td>det:predet</td>
<td>Predeterminer (all those ...)</td>
</tr>
</tbody>
</table>
Universal Dependencies
Nonverbal Predicate and Copula
1. Constituents vs. Dependencies

2. Universal Dependencies
   - A Tour through UD Syntax
   - Nonverbal Predicate and Copula
   - Core Arguments vs. Oblique Dependents
   - Ellipsis and Enhanced UD
Nonverbal Predicate and Copula

• Some languages use a copula verb:

Ivan is the best dancer.

• Some languages use a copula pronoun:

Ivan – to najlepszy tancerz.
Ivan – it best dancer.
Some languages use a copula verb:

Ivan is the best dancer.

Some languages omit the copula:

Иван лучший танцор.
Nonverbal Predicate and Copula

- Some languages use a copula verb:

  Ivan was the best dancer.

- Some languages use it only in some tenses:

  Ivan был лучшим танцором.
  Ivan was best dancer.
Copula Verbs: We Are Restrictive!

- *To be* is copula:

```
Ivan is the best dancer .
```

- *To become* is not copula:

```
Ivan became the best dancer .
```
Once Copula, Always Copula!

- This is parallel with Russian:

  
  \[
  \text{Ivan is the best dancer}. 
  \]

- This is also parallel with Russian:

  
  \[
  \text{Ivan is today in Moscow}. 
  \]
Well, Almost...

- This is parallel with Russian:

  Ivan is today in Moscow.

- But not with this in English:

  There is a dancer in Moscow.
Clauses and Copula

• A clause can be the subject:

```
The problem is that he is missing .
```

• Or it can be annotated as the nonverbal predicate (note the two subjects):

```
The problem is that he is missing .
```
Universal Dependencies

Core Arguments vs. Oblique Dependents
Outline

1. Constituents vs. Dependencies

2. Universal Dependencies
   - A Tour through UD Syntax
   - Nonverbal Predicate and Copula
   - Core Arguments vs. Oblique Dependents
   - Ellipsis and Enhanced UD
## Dependents of Clauses (Verbal or Not)

<table>
<thead>
<tr>
<th>Core</th>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsubj</td>
<td>csubj</td>
<td>advmod</td>
<td>aux</td>
<td></td>
</tr>
<tr>
<td>obl</td>
<td>advcl</td>
<td>discourse</td>
<td>cop</td>
<td></td>
</tr>
<tr>
<td>vocative</td>
<td></td>
<td></td>
<td>mark</td>
<td></td>
</tr>
<tr>
<td>dislocated</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>expl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Core</th>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>obl</td>
<td></td>
<td>advcl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expl</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Dependents of Verbs, Adjectives and Adverbs

<table>
<thead>
<tr>
<th>Core</th>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>obj</td>
<td>ccomp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>iobj</td>
<td>xcomp</td>
<td>advmod</td>
<td></td>
</tr>
<tr>
<td>obl</td>
<td>advcl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expl</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Non-Core</th>
<th>Nominal</th>
<th>Clausal</th>
</tr>
</thead>
<tbody>
<tr>
<td>obl</td>
<td></td>
<td></td>
</tr>
<tr>
<td>expl</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Dependents of Nominals

<table>
<thead>
<tr>
<th>Nominal</th>
<th>Clausal</th>
<th>Modifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nmod</td>
<td>acl</td>
<td>amod</td>
<td>det</td>
</tr>
<tr>
<td>appos</td>
<td></td>
<td>nummod</td>
<td>case</td>
</tr>
<tr>
<td>acl</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Constituents vs. Dependencies

- Universal Dependencies

- 48/110
Information Packaging

Constituents vs. Dependencies
Information Packaging

Constituents vs. Dependencies
He loaded the wagon with hay.
He loaded the wagon with hay

He loaded hay on the wagon
UD is NOT about Semantic Roles!

Constituents vs. Dependencies
The secret to understanding the design and current success of UD is to realize that the design is a very subtle compromise between approximately 6 things:

1. UD must be satisfactory on linguistic analysis grounds for individual languages.
2. UD must be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
3. UD must be suitable for rapid, consistent annotation by a human annotator.
4. UD must be easily comprehended and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing. ... it leads us to favor traditional grammar notions and terminology.
5. UD must be suitable for computer parsing with high accuracy.
6. UD must support well downstream language understanding tasks (relation extraction, reading comprehension, machine translation, ...)

It’s easy to come up with a proposal that improves UD on one of these dimensions. The interesting and difficult part is to improve UD while remaining sensitive to all these dimensions.
UD Avoids Argument-Adjunct Distinction!

Constituents vs. Dependencies
Avoiding an Argument-Adjunct Distinction

- From the guidelines:
  - Subtle, unclear, and frequently argued over
  - Questionable as a categorical distinction
  - Best practical solution is to eliminate it

BUT:
- Cannot be eliminated completely
- Some people/data have it and want to keep it
- It aligns well with traditional grammars

⇒ there is now a relation subtype obl:arg

AND
- I will argue that Core-oblique distinction is unclear and argued over too
- (Though I will not propose to discard it.)
Avoiding an Argument-Adjunct Distinction

• From the guidelines:
  • Subtle, unclear, and frequently argued over
  • Questionable as a categorical distinction
  • Best practical solution is to eliminate it

• BUT:
  • Cannot be eliminated completely
  • Some people/data have it and want to keep it
    • It aligns well with traditional grammars
  • ⇒ there is now a relation subtype ob1:arg
Avoiding an Argument-Adjunct Distinction

• From the guidelines:
  • Subtle, unclear, and frequently argued over
  • Questionable as a categorical distinction
  • Best practical solution is to eliminate it

• BUT:
  • Cannot be eliminated completely
  • Some people/data have it and want to keep it
    • It aligns well with traditional grammars
  • ⇒ there is now a relation subtype obl:arg

• AND I will argue that
  • Core-oblique distinction is unclear and argued over too
  • (Though I will not propose to discard it.)
So What Is Core and Why?

Constituents vs. Dependencies
Universal Dependencies
Community Confusion

• UD v1 guidelines took core-oblique for granted

• English (simplified):
  • Bare noun phrase $\Rightarrow$ core argument ($\text{nsubj}$, $\text{obj}$, $\text{iobj}$)
  • Prepositional phrase $\Rightarrow$ oblique argument or adjunct ($\text{obl}$)
Community Confusion

- UD v1 guidelines took core-oblique for granted

- English (simplified):
  - Bare noun phrase \(\Rightarrow\) core argument (\text{nsubj, obj, iobj})
  - Prepositional phrase \(\Rightarrow\) oblique argument or adjunct (\text{obl})

- Other languages: not necessarily! (Spanish, Japanese)
  - But some people simply took the English rule...
  - Manning’s law: non-linguists!
Community Confusion

- UD v1 guidelines took core-oblique for granted

- English (simplified):
  - Bare noun phrase $\Rightarrow$ core argument (nsubj, obj, iobj)
  - Prepositional phrase $\Rightarrow$ oblique argument or adjunct (obl)

- Other languages: not necessarily! (Spanish, Japanese)
  - But some people simply took the English rule...
  - Manning’s law: non-linguists!

- Clash with traditional terminology
  - Grammars of German, Czech etc. define prepositional objects
  - But these are not necessarily core...
  - Yet some people took their national definition of object...
Language-specific Coding Strategy

• Idea:
  • Oblique arguments are marked similarly to adjuncts (prepositions, certain morphological cases...)
  • Core arguments are marked differently
    • ⇒ easy for annotators and non-linguists!

• Why are core arguments special?
  • They tend to be targeted by grammatical rules
    • Passivization
    • Control verbs
    • Reflexives
    • ...

Constituents vs. Dependencies

57/110
Language-specific Coding Strategy

- Core vs. oblique is not defined in traditional grammar
- How shall we define it?

Andrews, 2007 (In Shopen: Language Typology)

- Identify primary transitive predicates
- We need semantic roles for this! (One-time only.)

  - Actor/agent = function $A$
  - Undergoer/patient = function $P$

  - Note the way they are coded
  - Note other grammatical rules that target them
  - Generalize to other predicates with same coding and rules
  - Then define:
    - function $A \Rightarrow \text{nsubj}$
    - function $P \Rightarrow \text{obj}$
Language-specific Coding Strategy

- Core vs. oblique is not defined in traditional grammar
- How shall we define it?

- Andrews, 2007 (In Shopen: Language Typology)
  - Identify primary transitive predicates
    - We need semantic roles for this! (One-time only.)

  - Identify primary transitive predicates
    - We need semantic roles for this! (One-time only.)
• Core vs. oblique is not defined in traditional grammar
• How shall we define it?

• Andrews, 2007 (In Shopen: Language Typology)
  • Identify primary transitive predicates
  • We need semantic roles for this! (One-time only.)
  • Actor/agent = function \( A \)
  • Undergoer/patient = function \( P \)
Language-specific Coding Strategy

- Core vs. oblique is not defined in traditional grammar
- How shall we define it?

- Andrews, 2007 (In Shopen: Language Typology)
  - Identify *primary transitive predicates*
    - We need semantic roles for this! (One-time only.)
  - Actor/agent = function $A$
  - Undergoer/patient = function $P$
  - Note the way they are coded
  - Note other grammatical rules that target them
  - Generalize to other predicates with same coding and rules
Language-specific Coding Strategy

• Core vs. oblique is not defined in traditional grammar
• How shall we define it?

• Andrews, 2007 (In Shopen: Language Typology)
  • Identify primary transitive predicates
    • We need semantic roles for this! (One-time only.)
  • Actor/agent = function $A$
  • Undergoer/patient = function $P$
  • Note the way they are coded
  • Note other grammatical rules that target them
  • Generalize to other predicates with same coding and rules

• Then define:
  • function $A \Rightarrow \text{nsubj}$
  • function $P \Rightarrow \text{obj}$
Transitive Predicates in English

John kills Mary (primary transitive)

PROPN VERB PROPN
Transitive Predicates in English

John kills Mary (primary transitive)

PROPN VERB PROPN

John loves Mary (generalized transitive)

PROPN VERB PROPN
Transitive Predicates in English

- John \( \rightarrow \text{kills} \rightarrow \text{Mary} \) (primary transitive)
  - Constituents: John, kills, Mary
  - Dependencies: nsubj (John), obj (Mary)

- John \( \rightarrow \text{loves} \rightarrow \text{Mary} \) (generalized transitive)
  - Constituents: John, loves, Mary
  - Dependencies: nsubj (John), obj (Mary)
Transitive Predicates in English

- **Case=Nom**
- **Voice=Act, Pass**
- **Case=Acc**

### Constituents vs. Dependencies

- **nsubj** → **nominal**
  - Case=Nom, bare NP, pre-verb, cross-ref on verb

- **obj** → **nominal**
  - Case=Acc, bare NP, post-verb

- **VERB**
  - Voice=Act, Pass, declarative clause

- **agreement** → **cross-ref on verb**
Passivization in English

Constituents vs. Dependencies
Subject Control in English

Constituents vs. Dependencies

John wants to kill Mary

John wants to love Mary

Universal Dependencies
Object Control in English

Constituents vs. Dependencies
Some Problems

- Some temporal adjuncts are bare noun phrases
  - I work the whole week.
  - I work every Friday.

- At least it cannot passivize:
  - *The whole week is worked by me.
  - *Every Friday is worked by me.

But...
Some Problems

- Some transitive verbs cannot passivize
  - John *has a new car.*
  - *A new car is had by John.*
  - Friday does not *suit* me.
  - *I am not suited by Friday.*
Some Problems

- Some transitive verbs cannot passivize
  - John has a new car.
  - *A new car is had by John.
  - Friday does not suit me.
  - *I am not suited by Friday.

- Some prepositional verbs can passivize
  - You can rely on Ben.
    - Ben can be relied on.
  - They will take care of your children.
    - Your children will be taken care of.
Tentative Summary?

- The borderline is inherently fuzzy
- No universally applicable and exact algorithm
- Better described in terms of probability
Tentative Summary?

• The borderline is inherently fuzzy
• No universally applicable and exact algorithm
• Better described in terms of probability

• Core coding not favored by adjuncts
• Oblique coding similar to most adjuncts
• Passivization etc. may help...
• ... but does not work as strict criterion
• The borderline is inherently fuzzy
• No universally applicable and exact algorithm
• Better described in terms of probability

• Core coding not favored by adjuncts
• Oblique coding similar to most adjuncts
• Passivization etc. may help…
• … but does not work as strict criterion

• Semantic roles needed when starting a new language
• Argument-adjunct needed to describe exceptions (*the whole week*)
Intransitive Predicates

• Just one core argument
  • We already “know” how to find out if there are two

• ⇒ function \( S \)
  • Regardless of semantic role:
    • \( John \) runs.
    • \( John \) sleeps.
    • \( John \) falls.

• Then define:
  • function \( S \) ⇒ nsubj
Ditransitive Predicates

- Three core arguments
- Is one of them “least core”? ⇒ iobj
- (Alternatively, we could look at the semantic roles once again.)

```
I gave her a book
```

- Passivization:
  - *She was given a book by me.*
  - *?A book was given her by me.*
• Three core arguments
• Is one of them “least core”? ⇒ iobj
• (Alternatively, we could look at the semantic roles once again.)

Andrews (2007): *the status of the notion of ‘indirect object’ is problematic and difficult to sort out. The top priority is to work out what properties recipients and themes do and do not share with P arguments of primary transitive verbs.*
Jorge mató al dragón

El dragón fue matado por Jorge
Spanish Transitive Clauses

(a) nominal
   Case=Nom
   bare NP
   pre-verb
   cross-ref on verb
   ... nominal
   Case=Acc
   (or bare NP)
   post-verb

VERB
   Voice=Act(,Pass)
   declarative clause
   ← agreement

root

obj

case

constituents vs. dependencies
Spanish Adjunct Exceptions

- El trabaja toda la semana
  - He works whole the week
  - **PRON** **VERB** **DET** **DET** **NOUN**

- Subiremos a el tren a las cinco
  - We will board to the train at the five
  - **VERB** **ADP** **DET** **NOUN** **ADP** **DET** **NUM**

Constituents vs. Dependencies

Universal Dependencies
Spanish Ditransitive Clauses

Constituents vs. Dependencies
Spanish Ditransitive Clauses

Pedro le dio un libro a Isabel

PROPN PRON VERB DET NOUN ADP PROPN

Un libro fue dado a Isabel por Pedro

DET NOUN AUX VERB ADP PROPN ADP PROPN

Constituents vs. Dependencies
Jiří zabil draka.

Drak byl zabit Jiřím.
Czech Transitive Clauses

Constituents vs. Dependencies

Universal Dependencies

73/110
Czech Adjunct Exceptions

Pracuje celý týden
He-works whole week

VERB ADJ NOUN

Case=Acc
Czech Ditransitive Clauses

Constituents vs. Dependencies
Dative: Recipient vs. Beneficiary

Constituents vs. Dependencies

PROPN Petr
VERB četl
NOUN knihu

PROPN Petr
VERB zlomil
NOUN nohu

PROPN Katce
VERB to-Katka
NOUN book

PROPN Katce
VERB zlomil
NOUN nohu
Monotransitive with Dative?

Zuzka pomohla Martinovi úkolem

PROPN VERB PROPN ADP NOUN
Case=Nom Case=Dat Case=Ins

Martinovi bylo pomoženo s úkolem

PROPN AUX ADP NOUN
Case=Dat Gender=Neut Number=Sing Case=Ins
Monotransitive with Genitive?

Novináři musí dbát zásad objektivity.

Musí být dbáno zásad objektivity.
Monotransitive with Instrumental?

Karel hýbal nábytkem
Karel moved furniture

Nábytkem bylo hýbáno
Furniture was moved

Constituents vs. Dependencies
Spoléhali na ředitelovo rozhodnutí
They relied on director’s decision

Na ředitelovo rozhodnutí bylo spoléháno
On director’s decision was relied

Constituents vs. Dependencies
• There is a core-oblique scale:
  - Nom > Acc > Gen, Dat > Ins > preposition

• Where is the borderline?

⇒ No ditransitives in Czech!
(Exception: učit “to teach” takes two Acc.)
There is a core-oblique scale:

- **Nom > Acc > Gen, Dat > Ins > preposition**

Where is the borderline?

- **UD Czech 1.0: object = argument**
  - Nom, Acc, Gen, Dat, Ins, ADP > “adverbial”

⇒ No ditransitives in Czech!

(Exception: **učit** “to teach” takes two Acc.)
There is a core-oblique scale:
- **Nom > Acc > Gen, Dat > Ins > preposition**

Where is the borderline?

- UD Czech 1.0: object = argument
  - Nom, Acc, Gen, Dat, Ins, ADP > “adverbial”
- UD Czech 2.1–2.5: bare NP > PP
  - Nom, Acc, Gen, Dat, Ins > ADP + adjuncts
There is a core-oblique scale:

- **Nom > Acc > Gen, Dat > Ins > preposition**

Where is the borderline?

- UD Czech 1.0: object = argument
  - Nom, Acc, Gen, Dat, Ins, ADP > "adverbial"
- UD Czech 2.1–2.5: bare NP > PP
  - Nom, Acc, Gen, Dat, Ins > ADP + adjuncts
- UD Czech 2.6 (May 2020):
  - Nom, Acc > Gen, Dat, Ins, ADP + adjuncts
• There is a core-oblique scale:
  - Nom > Acc > Gen, Dat > Ins > preposition

• Where is the borderline?

• UD Czech 1.0: object = argument
  - Nom, Acc, Gen, Dat, Ins, ADP > “adverbial”

• UD Czech 2.1–2.5: bare NP > PP
  - Nom, Acc, Gen, Dat, Ins > ADP + adjuncts

• UD Czech 2.6 (May 2020):
  - Nom, Acc > Gen, Dat, Ins, ADP + adjuncts
  - ⇒ No ditransitives in Czech!
  - (Exception: učit “to teach” takes two Acc.)
Basque Transitive Clauses

Constituents vs. Dependencies
Basque Transitive Clauses

Constituents vs. Dependencies

83/110
Basque Intransitive Clauses

- **Gizona hilda**
  - Case=Abs
  - Noun: The-man
  - Verb: died
  - Aux: it-has

- **Urak irakin du**
  - Case=Erg
  - Noun: Water
  - Verb: boiled
  - Aux: it-has-it

Constituents vs. Dependencies
Basque Ditransitive Clauses

 Constituents vs. Dependencies
Basque Ditransitive Clauses

\[
\begin{align*}
\text{Iñakiki} & \quad \text{liburua} & \quad \text{eman} & \quad \text{zion} & \quad \text{Arantxari} \\
\text{PROPN} & \quad \text{NOUN} & \quad \text{VERB} & \quad \text{AUX} & \quad \text{PROPN} \\
\text{Case=Erg} & \quad \text{Case=Abs} & \quad & \quad \text{Case=Dat} \\
\text{nsubj} & \quad \text{obj} & \quad \text{aux} & \quad \text{iobj} & \\
\text{root} & \\
\end{align*}
\]

\[
\begin{align*}
\text{Zezenak} & \quad \text{saihetsa} & \quad \text{pitzątu} & \quad \text{zion} & \quad \text{Iñakiri} \\
\text{Bull} & \quad \text{rib} & \quad \text{cracked} & \quad \text{it-has-it-it} & \quad \text{to-Iñaki} \\
\text{NOUN} & \quad \text{NOUN} & \quad \text{VERB} & \quad \text{AUX} & \quad \text{PROPN} \\
\text{Case=Erg} & \quad \text{Case=Abs} & \quad \text{Case=Dat} & \\
\text{nsubj} & \quad \text{obj} & \quad \text{aux} & \quad \text{iobj} & \\
\text{root} & \\
\end{align*}
\]
Basque Causative Applied to Dative Subject

Constituents vs. Dependencies

- **Zopa**
  - **izugarri**
  - **gustatzen**
  - **zaio**
  - **mutilari**
  - **Soup**
  - **greatly**
  - **pleasing**
  - **it-is-it**
  - **to-boy**
  - **Case=Abs**
  - **Voice=Act**
  - **Case=Dat**

- **Goseak**
  - **izugarri**
  - **gustatuerazi**
  - **zion**
  - **mutilari**
  - **Hunger**
  - **soup**
  - **greatly**
  - **made-pleasing**
  - **it-has-it-it**
  - **to-boy**
  - **Case=Erg**
  - **Case=Abs**
  - **Voice=Cau**
  - **Case=Dat**
Yidiɲ Transitive Clauses

Constituents vs. Dependencies
Yidiŋ “Dative” Adnominal Clauses

“І, (who) was slapped by the woman, laughed”

Case=Nom
PRON
ŋayu
I

Case=Acc
VERB
mangaːŋ
laughed

Case=Erg
PRON
(ŋaɲaɲ)
me

Case=Erg
NOUN
bũnaːŋ
woman

Case=Acc
VERB
wuɾaːɲunda
slapping
Yidiŋ “Dative” Adnominal Clauses

“I, (who) was slapped by the woman, laughed”

The coreferential (and elidable) NP must have S or P function.
“I, (who) was laughing, was slapped by the woman”

The coreferential (and elidable) NP must have S or P function.
“I, (who) was slapping the woman, laughed”

Original P is now oblique and original A is now S.
Tagalog Transitive Clauses

Constituents vs. Dependencies
Magaalis will-take ang babae ng bigas sa sako.

Voice=Act? Case=Nom

Aalisin will-take ng babae ang bigas sa sako.

Voice=Pass? Case=Gen
Tagalog Locative Voice ⇒ Ditransitive!

Aalisan ng babae ng sako
Verbs: Aalisan (Will-take)
Nouns: babae (woman), sako (sack)
Adpositions: ng (from-the)
Case: Gen (locative)
Voice: Lfoc (locative)

Dependencies:
- nsubj:lfoc
- obj:agent
- obj:patient
- det

Constituents vs. Dependencies
Tagalog Benefactive Voice ⇒ Ditransitive!

Ipagaalis ng babae ng bigas ang bata
Will-take woman rice for-the child

VERB NOUN NOUN ADP ADP NOUN
Voice=Bfoc Case=Gen Case=Gen Case=Nom

Constituents vs. Dependencies
Plains Cree Transitive Clauses

Constituents vs. Dependencies

96/110
Direct-Inverse Voice in Plains Cree

Animacy hierarchy: 1st person > 3rd person
Direct-Inverse Voice in Plains Cree

Animacy hierarchy: 1st person > 3rd person
Should we set \textbf{nsubj} > \textbf{obj}?
Direct-Inverse Voice in Plains Cree

Constituents vs. Dependencies
The **theme** (not the recipient) is indirect object because it is not cross-referenced on the verb (it is **inanimate**, while the verb references an animate object).
Universal Dependencies
Ellipsis and Enhanced UD
Outline

1. Constituents vs. Dependencies

2. Universal Dependencies
   - A Tour through UD Syntax
   - Nonverbal Predicate and Copula
   - Core Arguments vs. Oblique Dependents
   - Ellipsis and Enhanced UD
Deleted Predicates in Coordination

Kate went to Florida and Jane (went) to Europe

- Some treebanks would use an empty node to represent the second *went*.
- UD enhanced representation now allows empty nodes!
- But the basic representation sticks with the overt words.
Kate went to Florida and Jane (went) to Europe
PDT: The ExD Relation

Kate went to Florida and Jane (went) to Europe
Kate went to Florida and Jane (went) to Europe
UD V2: The orphan Relation

Kate went to Florida and Jane (went) to Europe
Enhanced Dependencies: Gapping

Kate wants to go to Florida and Jane (wants) (go) to Europe

Constituents vs. Dependencies
Kate wants to go to Florida and Jane (wants) (go) to Europe

Constituents vs. Dependencies

Enhanced Dependencies: Gapping and Control
Enhanced Dependencies: Coordination

Jane eats sweet apples and oranges

Constituents vs. Dependencies
A gdzie szukać szamponu, który myje?
And where to-look for-shampoo, that washes?
Basic Universal Dependencies: 138 (136) Languages and Growing

• I.-E.:  Armenian (+West),  Greek (+Ancient),  Albanian, Gheg,  Hittite,  Breton, Irish, Manx,  Scottish,  Welsh,  Afrikaans,  Danish,  Dutch,  English, Faroese,  Frisian, German,  Gothic,  Icelandic,  Low Saxon,  Norwegian, Swedish,  Swiss German,  Catalan,  French,  Galician,  Italian,  Latin, Ligurian,  Neapolitan,  Old French,  Portuguese,  Romanian,  Spanish, Umbrian,  Belarusian,  Bulgarian,  Church Slavonic,  Croatian,  Czech,  Old Russian,  Polish,  Pomak,  Russian,  Serbian,  Slovak,  Slovenian,  Ukrainian, Upper Sorbian,  Latvian,  Lithuanian,  Kurmanji,  Persian,  Khunsari,  Nayini, Soi, Urdu,  Hindi,  Kangri,  Bhojpuri,  Bengali,  Marathi,  Sinhala,  Sanskrit • Dravidian:  Malayalam,  Tamil,  Telugu • Uralic:  Erzya,  Estonian,  Finnish,  Hungarian,  Karelian, Livvi,  Komi Permyak+Zyrian,  Moksha,  Sámi North+Skolt • Turkic:  Kazakh,  Old Turkish,  Tatar,  Turkish,  Uyghur,  Yakut • Buryat • Xibe • Korean • Japanese • Sino-T.:  Cantonese,  Classical Chinese,  Chinese • Tai-Kadai:  Thai • Aus.-As.:  Vietnamese • Austron.:  Indonesian,  Javanese,  Tagalog,  Cebuano • Pama-Nyu.:  Warlpiri • Chu.-Kam.:  Chukchi • Esk.-Al.:  Yupik • U.-Az.:  Nahuatl • Mayan:  Kiche • Arawakan:  Apurinã • Arawan:  Madi • Tupian:  Akuntsu, Guajajara,  Kaapor,  Karo,  Makurap,  Mundurukú,  Nheengatu,  Tupinambá,  Mbyá,  Guaraní,  Teko • M.-Je:  Xavante • Af.-As.:  Akkadian,  Amharic,  Arabic Standard+Levantine, Assyrian,  Beja,  Coptic,  Hebrew (+Ancient),  Maltese,  Zaar • Niger-Congo:
Summary

- Constituent (phrase) trees ... context-free grammar
- Dependency trees (or graphs)
  - Nonprojective dependencies
- Universal Dependencies
  - Unified annotation for all languages
    - Language-specific extensions
  - Content words higher than function words ... better parallelism
  - Clauses – nominals – modifier words
  - Core arguments vs. oblique dependents

https://ufal.cz/courses/npfl094