A Brief Introduction to Universal Dependencies & Endangered Languages

Büşra Marşan
busra.marsan@boun.edu.tr

SCAN ME

What is parsing? Why do we need it?

- We need to put structures over sentences to understand/make computers understand how the words in the sentence relate to one another.

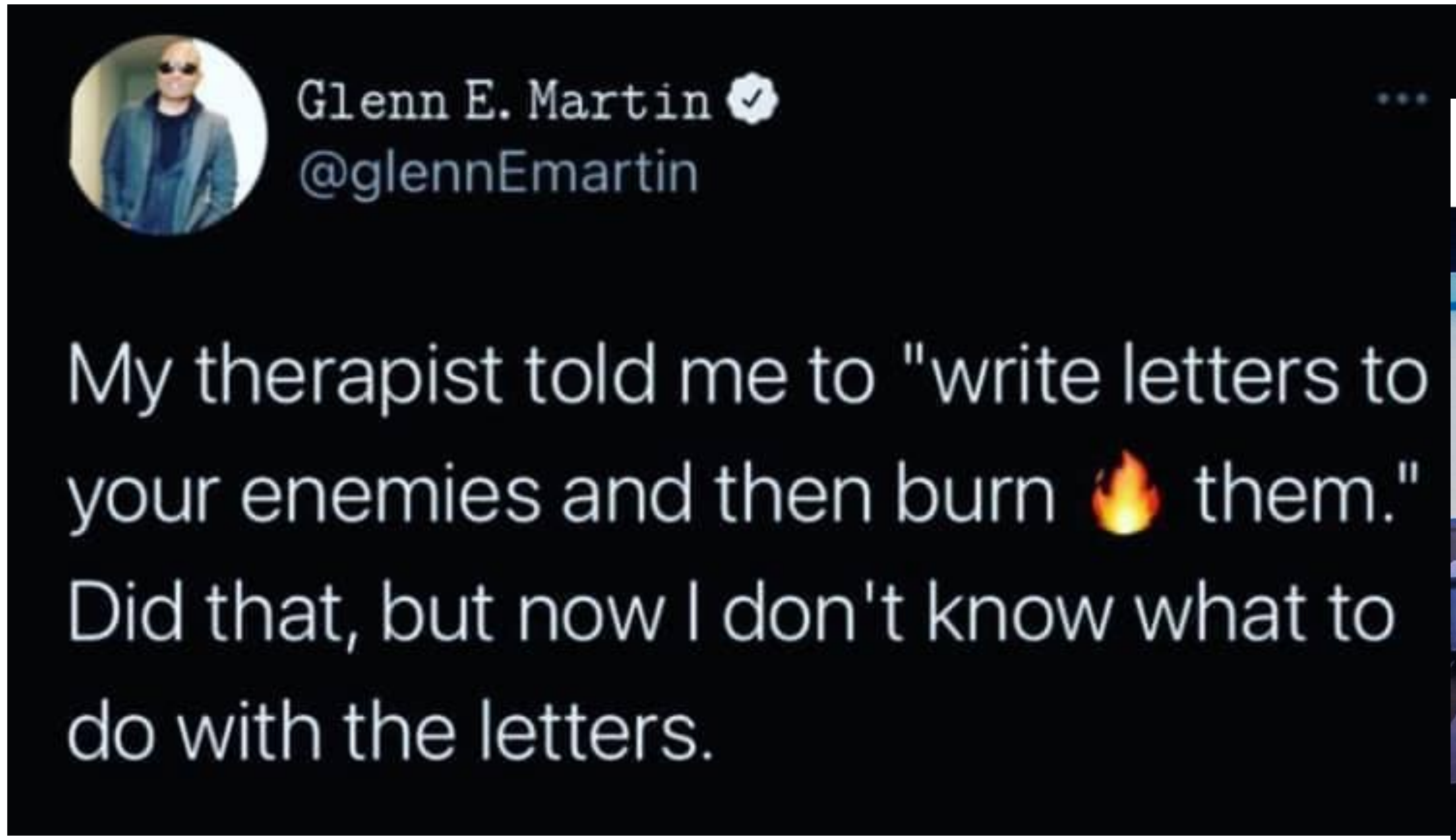
  The girl killed the bear. -vs- The bear killed the girl.

- So, parsing is practically determining the grammar structure of an input/sentence.

- A fancier definition: “*Analysing a text, made of a sequence of tokens (for example, words), to determine its grammatical structure within the framework of a (formal) grammar.”*

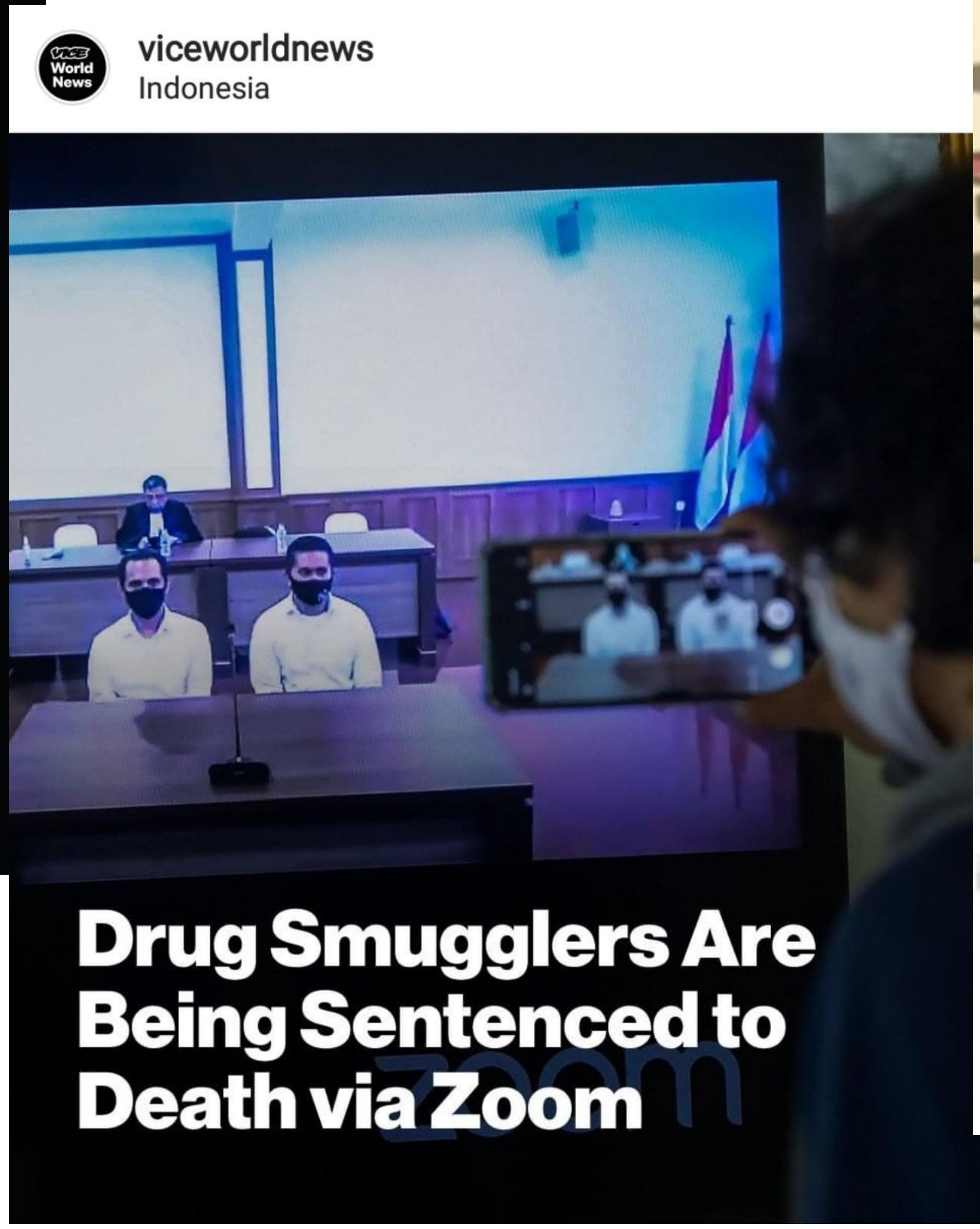
What is parsing? Why do we need it?

- Parsing is not a very easy task for computers since human languages can be VERY ambiguous. As a result computer scientists, computational linguists and scholars working in related fields came up with various frameworks and strategies.



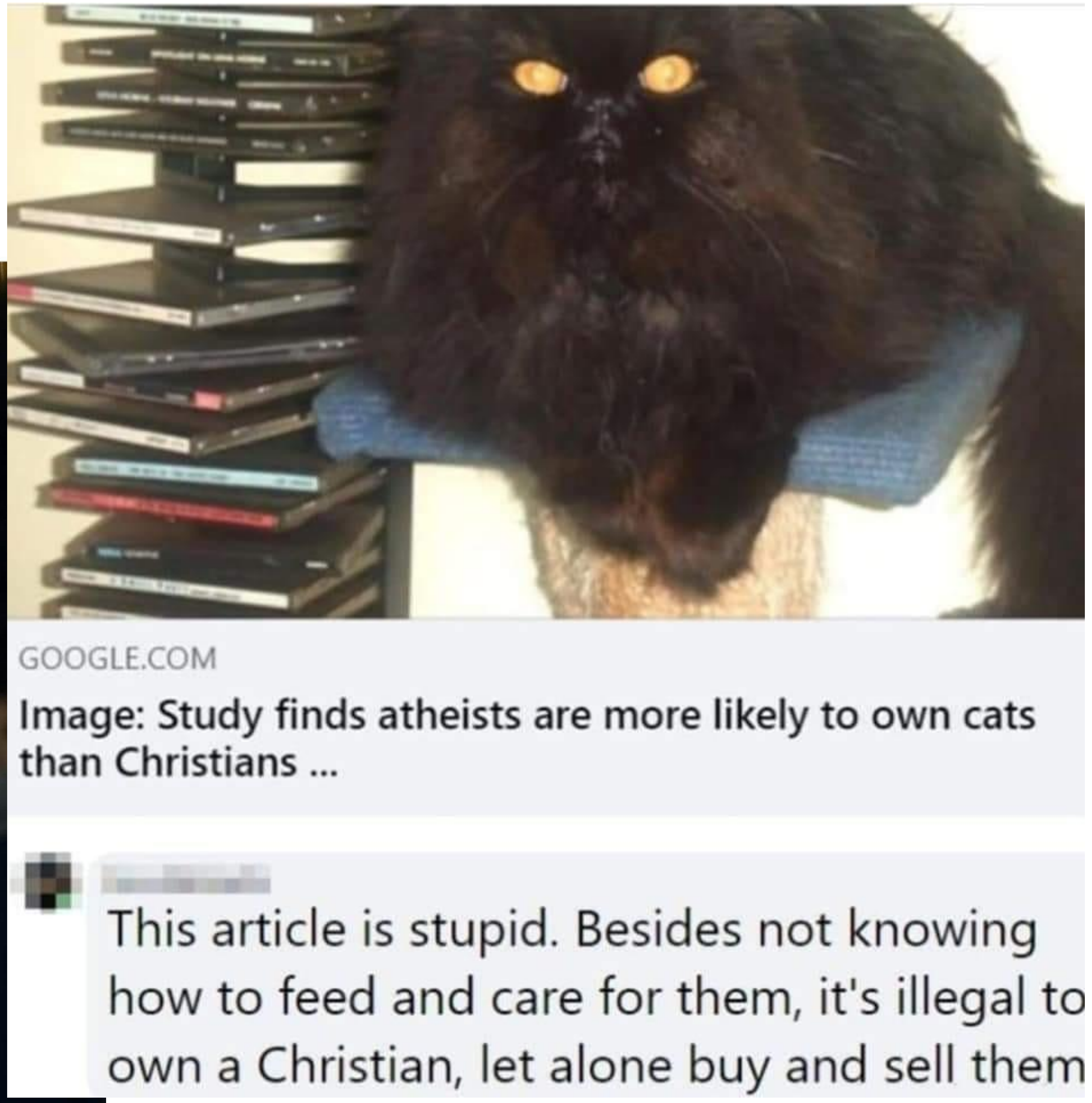
Glenn E. Martin
@glennEmartin

My therapist told me to "write letters to your enemies and then burn 🔥 them." Did that, but now I don't know what to do with the letters.



viceworldnews
Indonesia

Drug Smugglers Are Being Sentenced to Death via Zoom



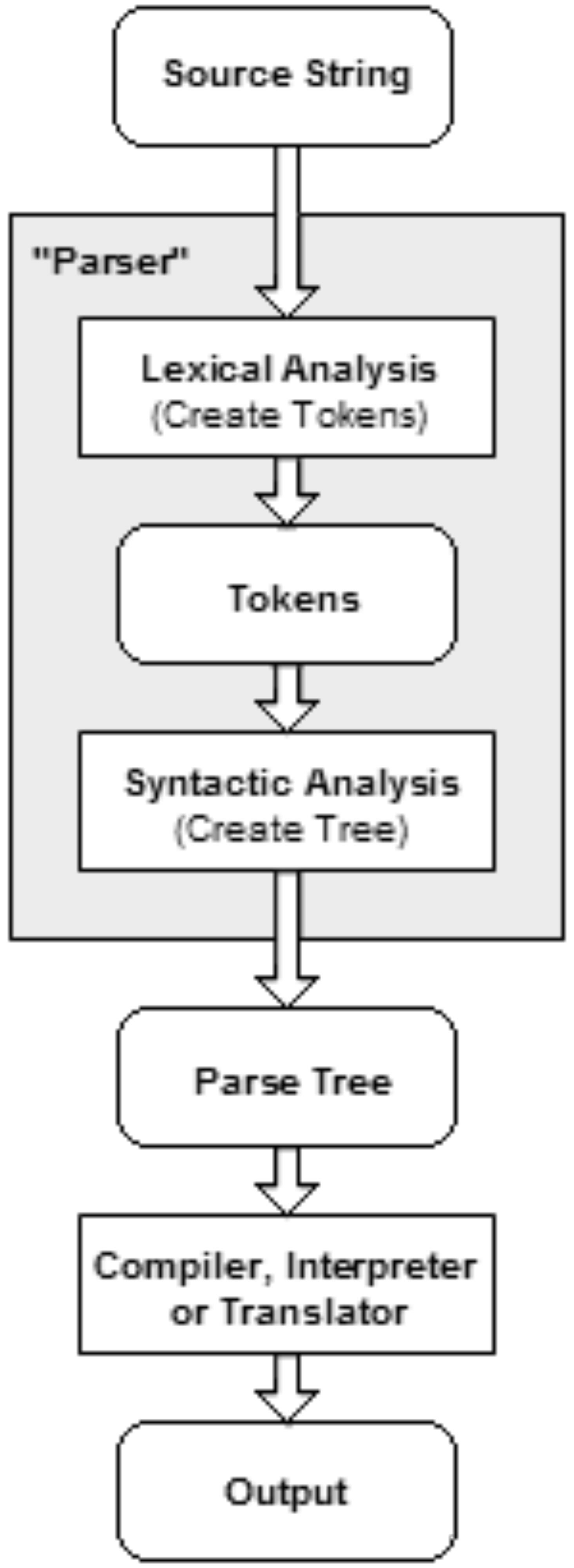
GOOGLE.COM

Image: Study finds atheists are more likely to own cats than Christians ...

[redacted]

This article is stupid. Besides not knowing how to feed and care for them, it's illegal to own a Christian, let alone buy and sell them

What is parsing? Why do we need it?



Source String

"Parser"

Lexical Analysis
(Create Tokens)

Tokens

Syntactic Analysis
(Create Tree)

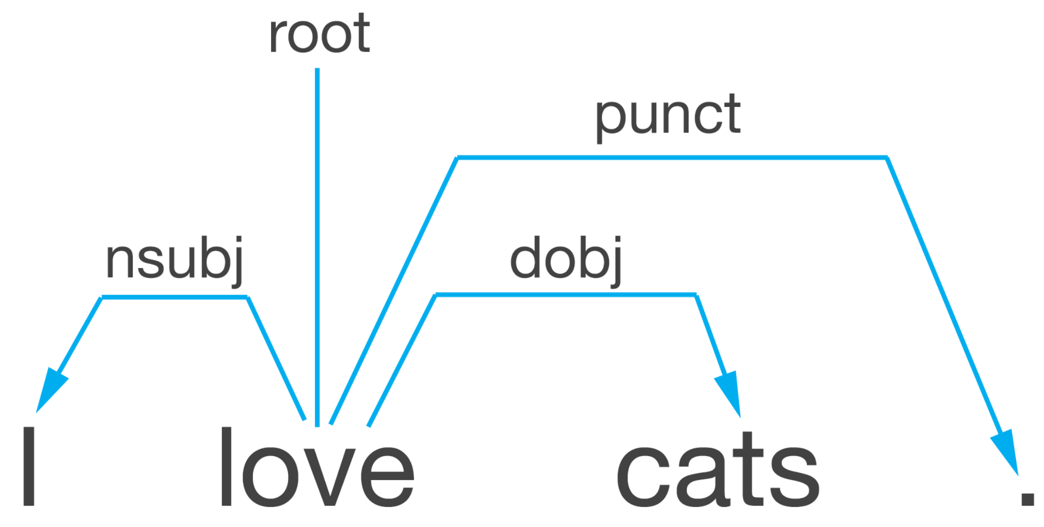
Parse Tree

Compiler, Interpreter
or Translator

Output

“I love cats.”

I, love, cats, .



Theoretical background on dependency grammars

Phrase structure grammar

- Chomsky
- Based on the notion of **constituency relations.**
    - Dates back to Aristotle & term logic.
    - Subject - predicate division.
- Binary branching and binary division. (X' Theory)

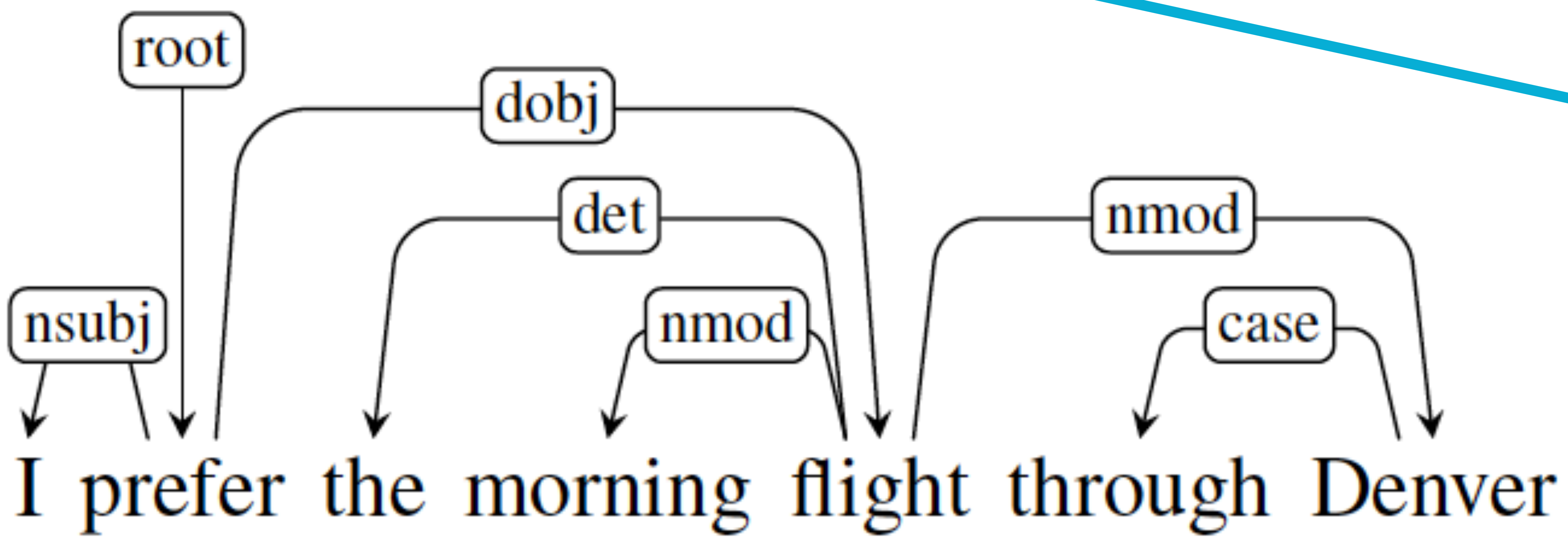
Theoretical background on dependency grammars

Dependency grammar

- Tesnière
- Based on the notion of **dependency relations.**
    - Defined using notions of head & dependent.
    - Linguistic units are connected to one another w/ links.
- Verb (predicate) is the king. Everything is connected to it directly or indirectly.
- Flatter. (no bar levels, no phrase levels etc.)

What is dependency parsing?

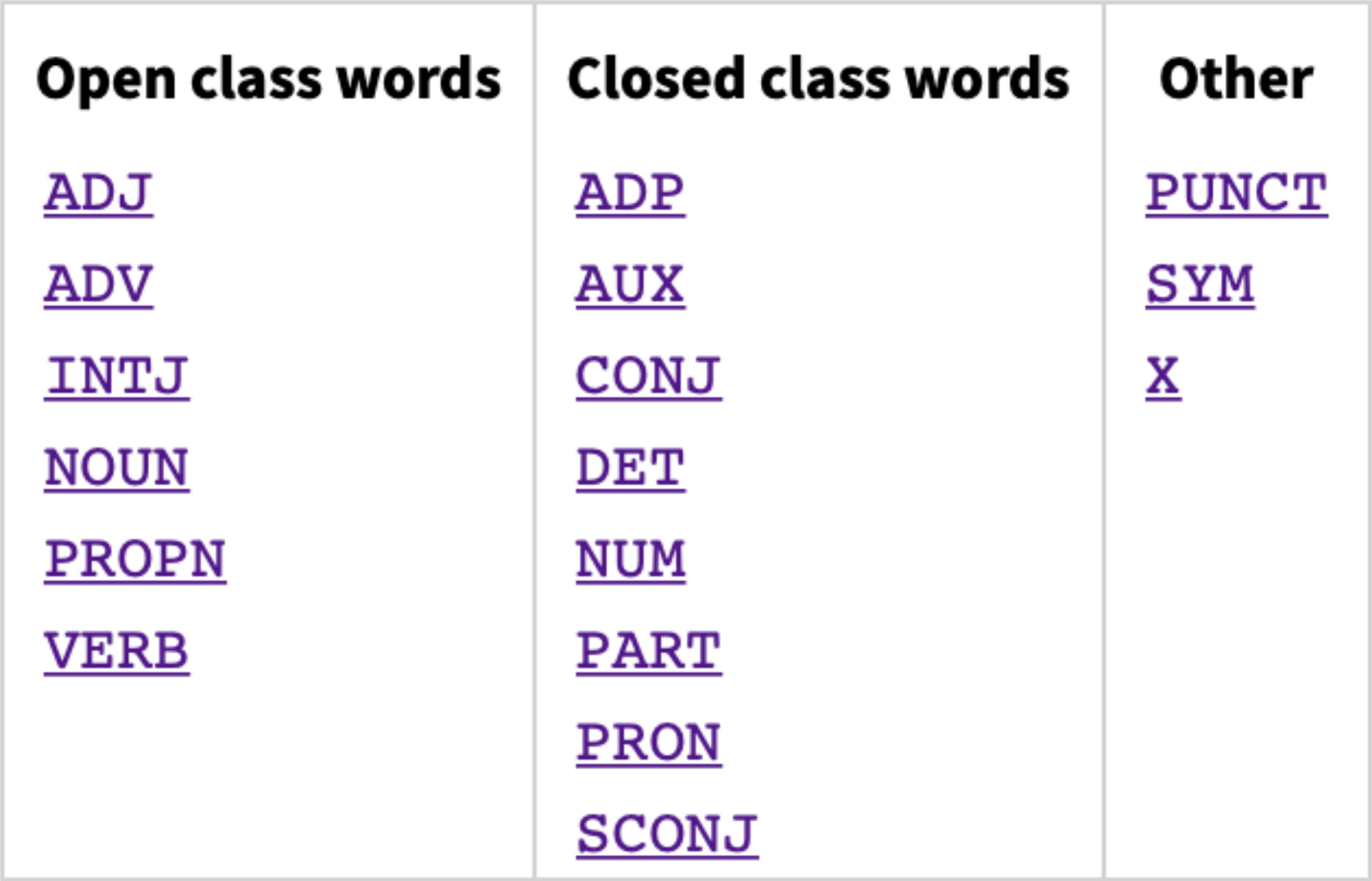
- Dependency parsing is based on dependency grammar.
- It illustrates relation between heads and their dependents using a set of **predefined tags.**



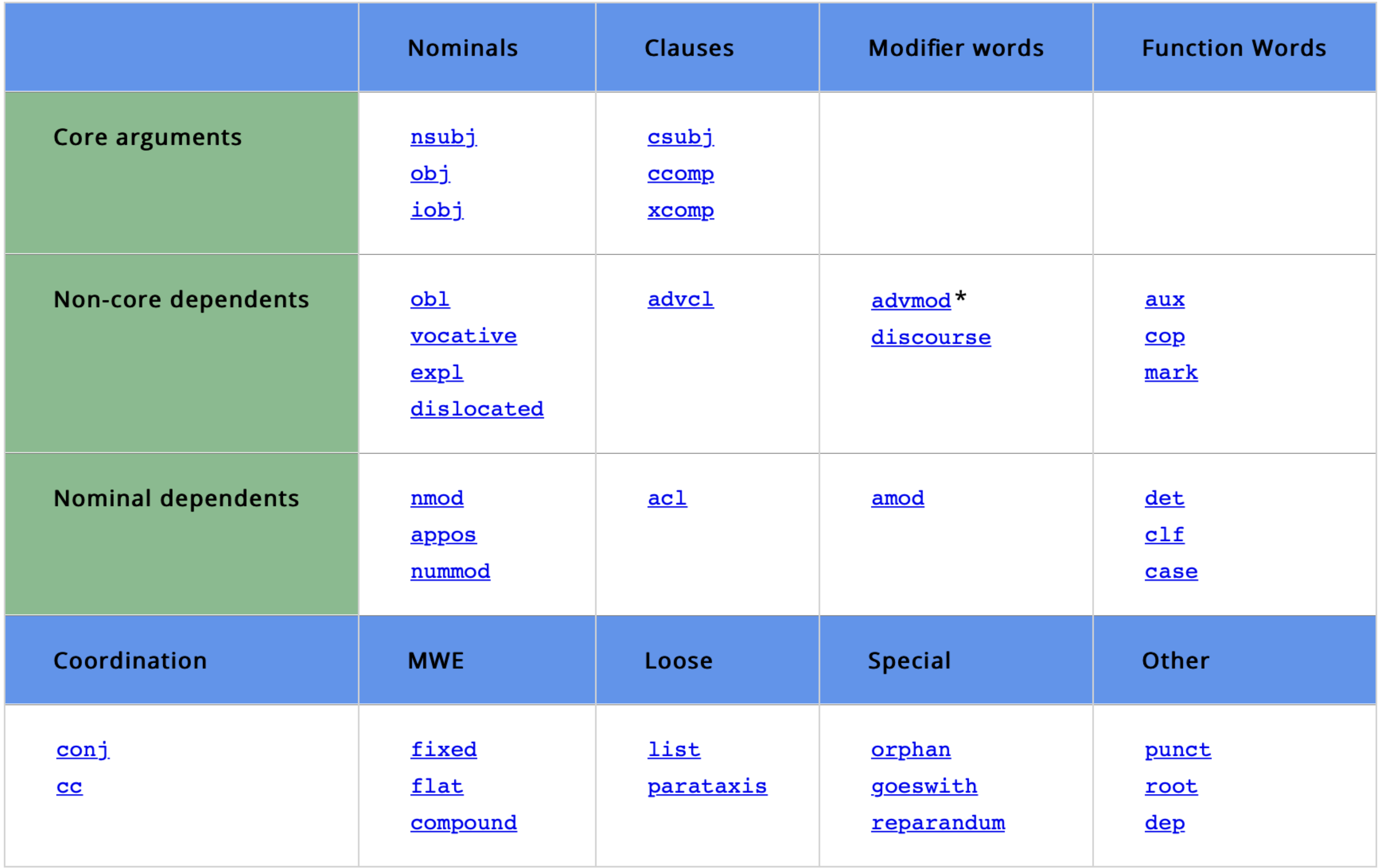
Mainly based on
POS tags

- Arrow implies head & its dependent, tag shows the relation.

POS Tags & Dependency Tags



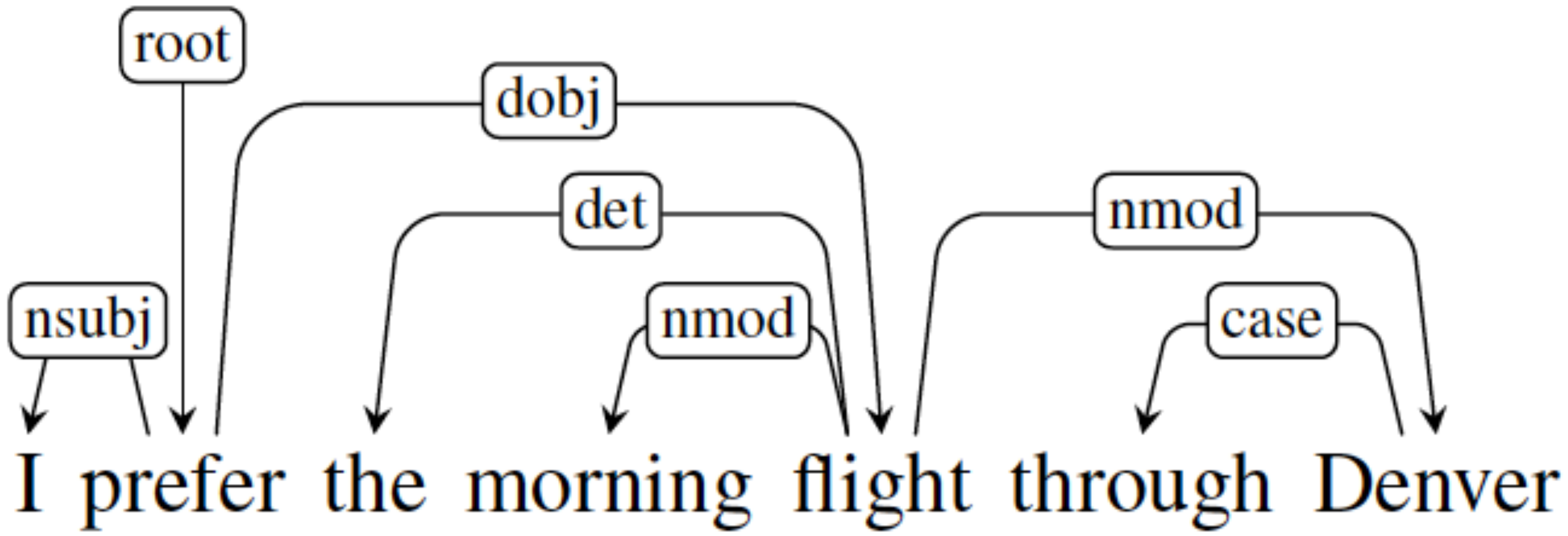
Open class words	Closed class words	Other
ADJ	ADP	PUNCT
ADV	AUX	SYM
INTJ	CONJ	X
NOUN	DET	
PROPN	NUM	
VERB	PART	
	PRON	
	SCONJ	



	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	MWE	Loose	Special	Other
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

Basic features of a dependency tree

- Each lemma has **<u>only one incoming arrow</u>**. Not zero, not two.
- Lemmas can have zero or multiple outgoing arrows.
- The predicate is the root. (“Lexical verb”)
- Function words cannot be heads. (Prepositions, articles, auxiliaries…)
- There must be a unique path between the root and each lemma.

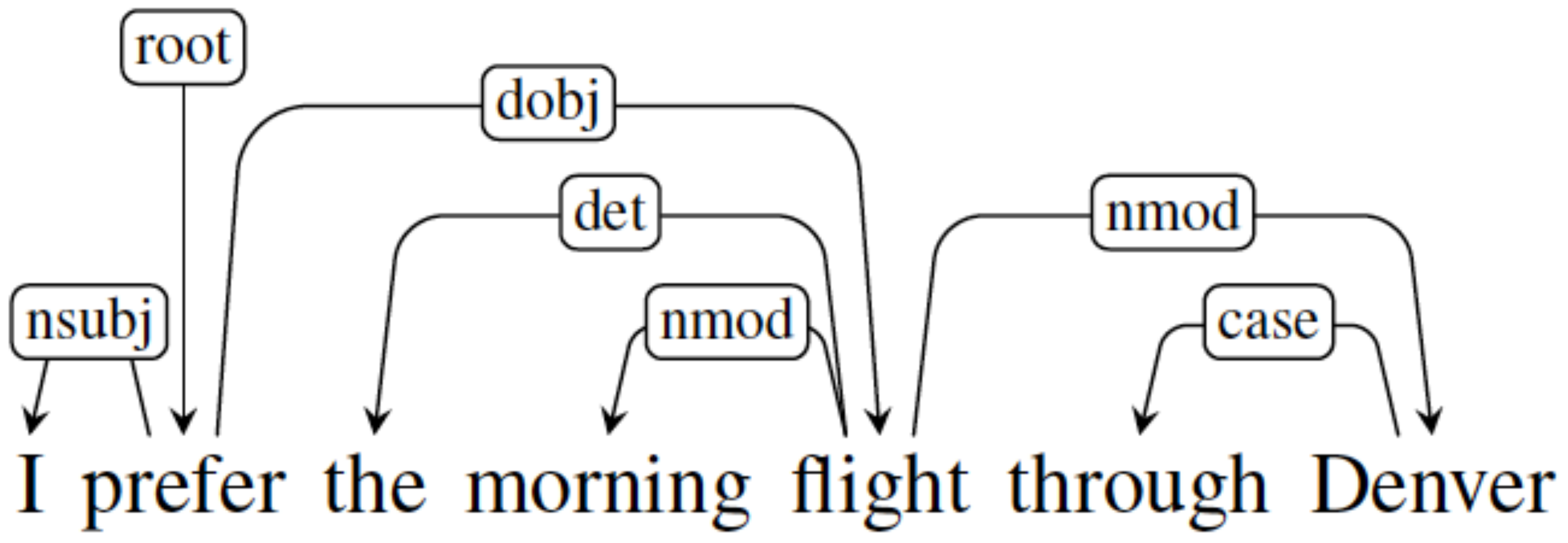


Some terminology

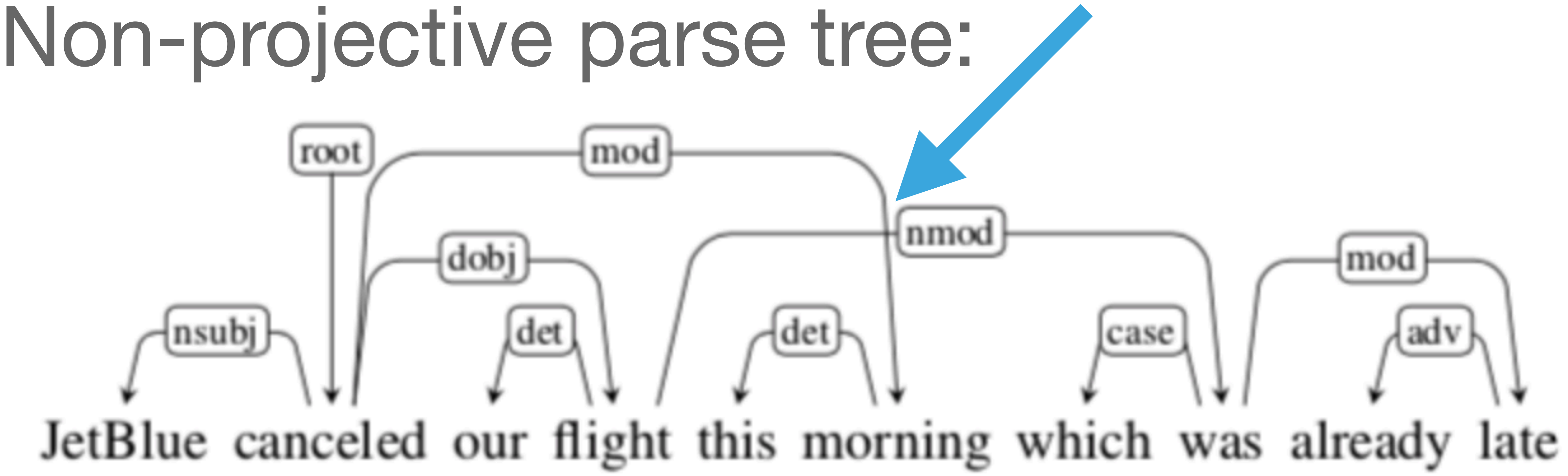
- **ROOT:** The root of the dependency tree.
- **LEMMA:** Lemma or stem of word form.
- **UPOSTAG:** Universal part-of-speech tag.
- **XPOSTAG:** Language-specific part-of-speech tag.

Some terminology

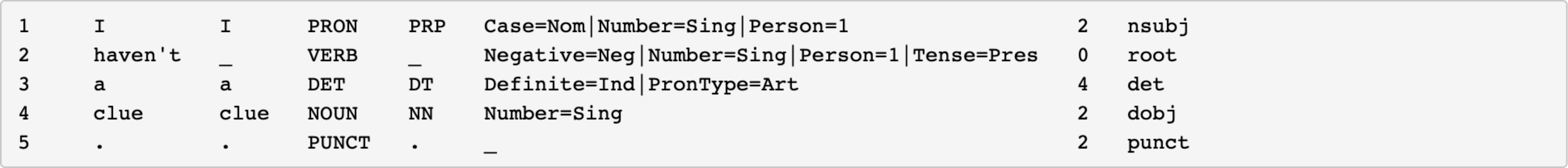
- Projective parse tree:



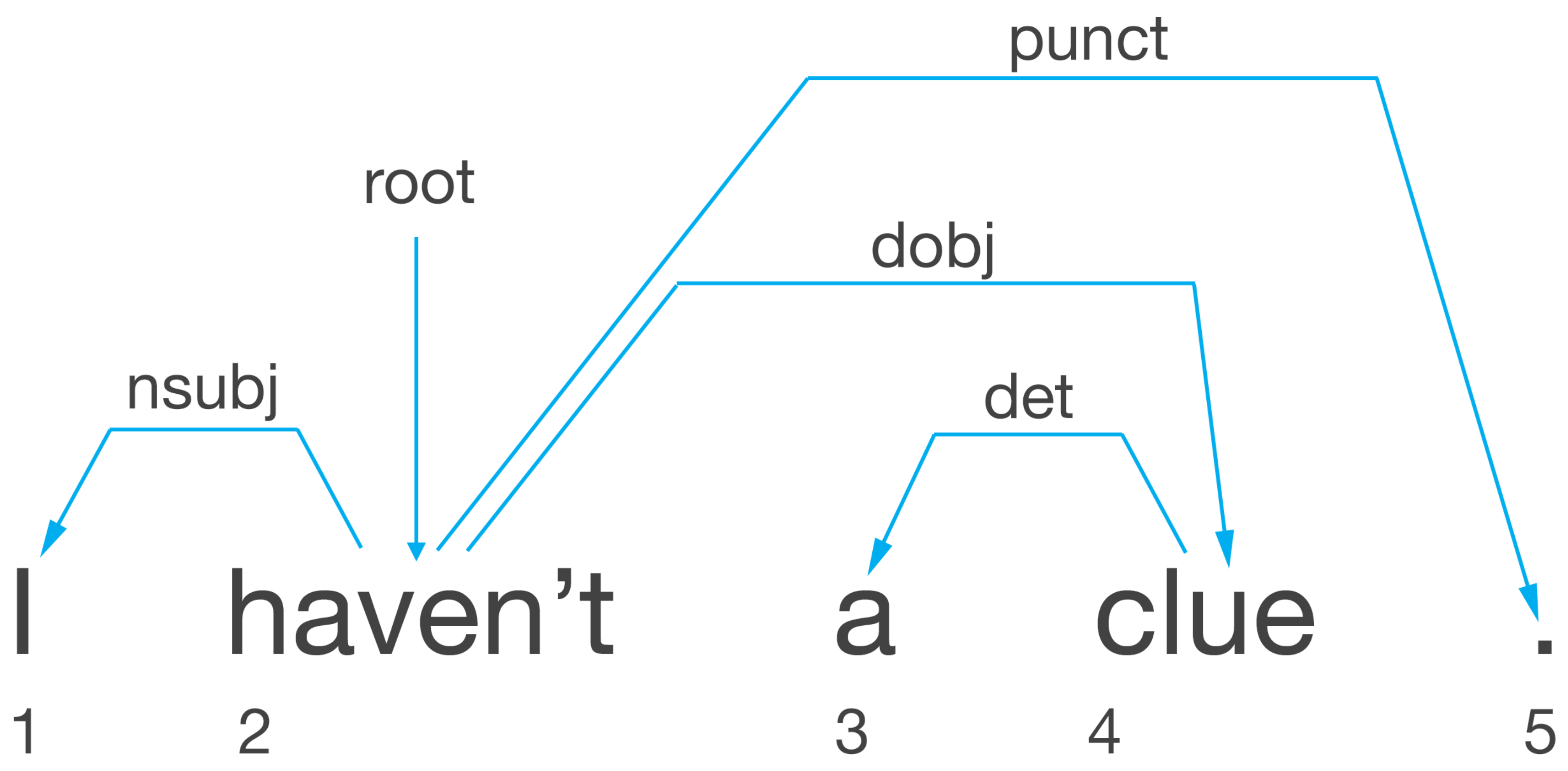
- Non-projective parse tree:


root
mod
nmod
dobj
mod
nsubj
det
det
case
adv
JetBlue canceled our flight this morning which was already late

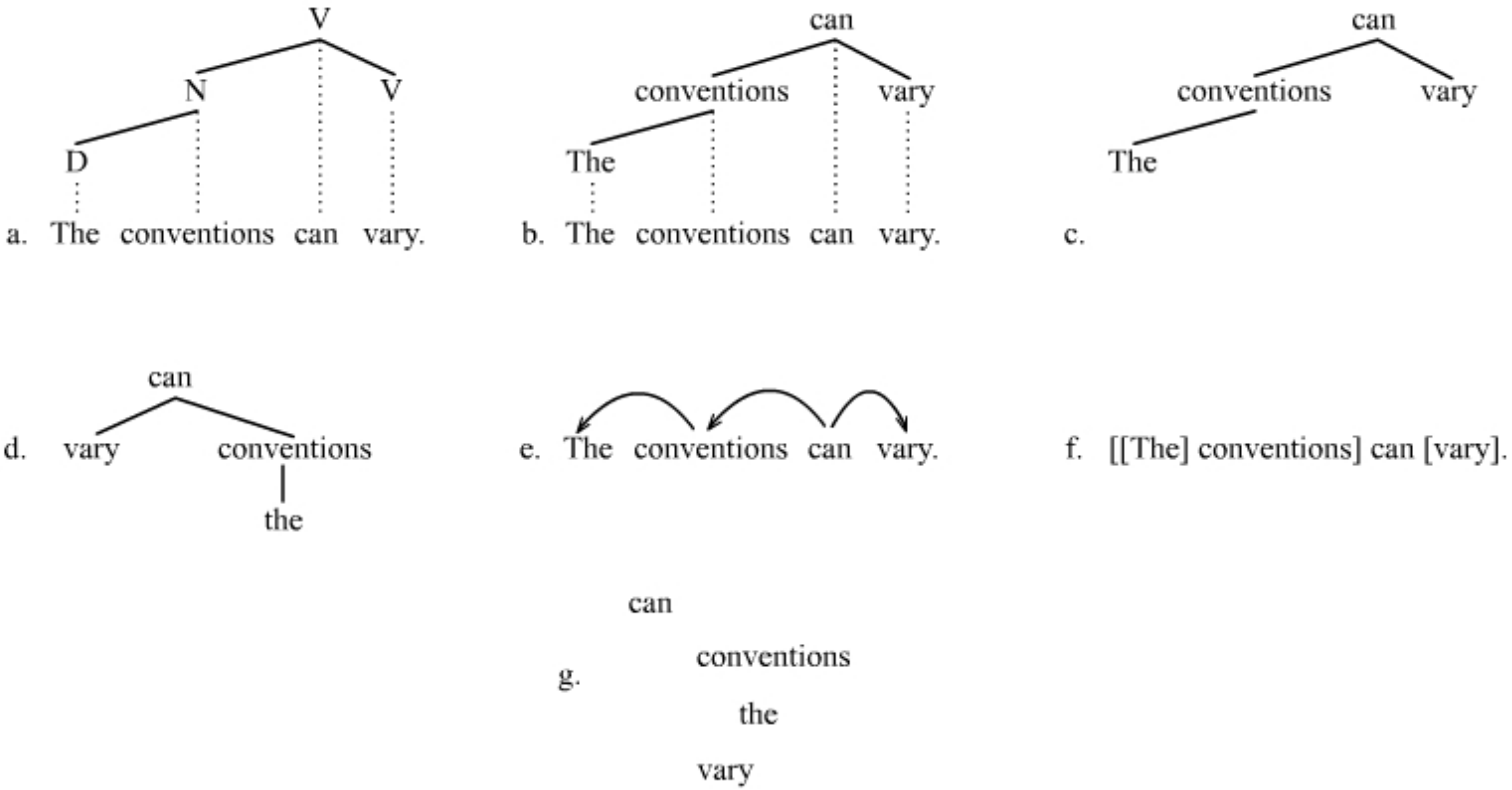
An actual dependency annotation



```
1    I        I      PRON   PRP   Case=Nom|Number=Sing|Person=1                   2    nsubj
2    haven't  _      VERB   _     Negative=Neg|Number=Sing|Person=1|Tense=Pres    0    root
3    a        a      DET    DT    Definite=Ind|PronType=Art                       4    det
4    clue     clue   NOUN   NN    Number=Sing                                     2    dobj
5    .        .      PUNCT  .     _                                               2    punct
```



Different visual representations of dependency trees


V
N
V
D
a. The conventions can vary.
can
conventions
vary
The
b. The conventions can vary.
can
conventions
vary
The
c.
can
d. vary
conventions
the
e. The conventions can vary.
f. [[The] conventions] can [vary].
can
conventions
g.
the
vary

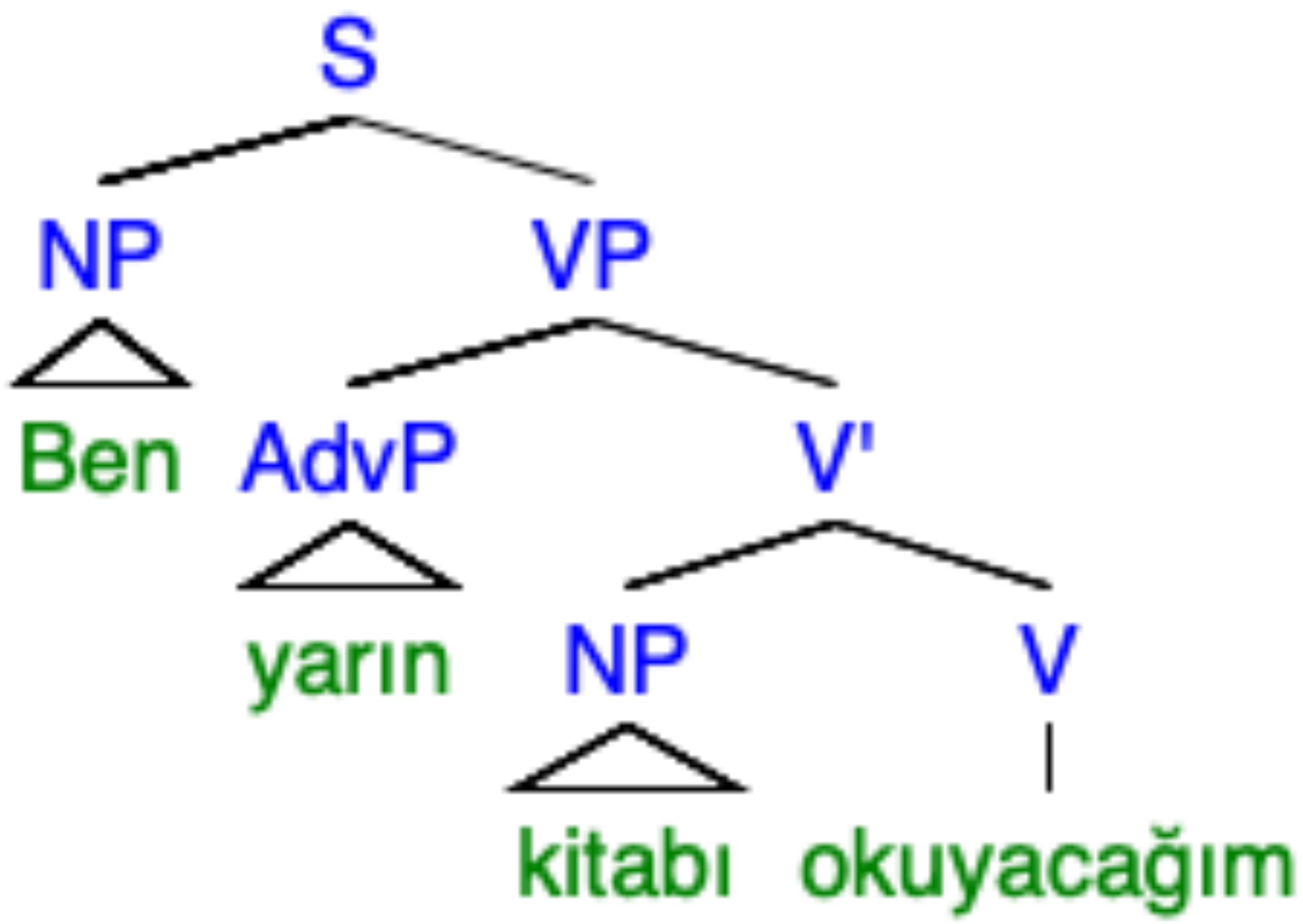
Dependency vs. Constituency trees

- Constituency trees were more popular in the past.
- Challenges for constituency trees: Projectivity, morphologically rich languages, free word order languages etc.
- Dependency trees are able to handle such challenges better. Famously, dependency models fit free word order languages much better.
- For a thorough discussion of dependency & constituency parsing in terms of parser performance: https://www.aclweb.org/anthology/P04-1061.pdf

Dependency vs. Constituency trees

“Kitabı yarın ben okuyacağım.”

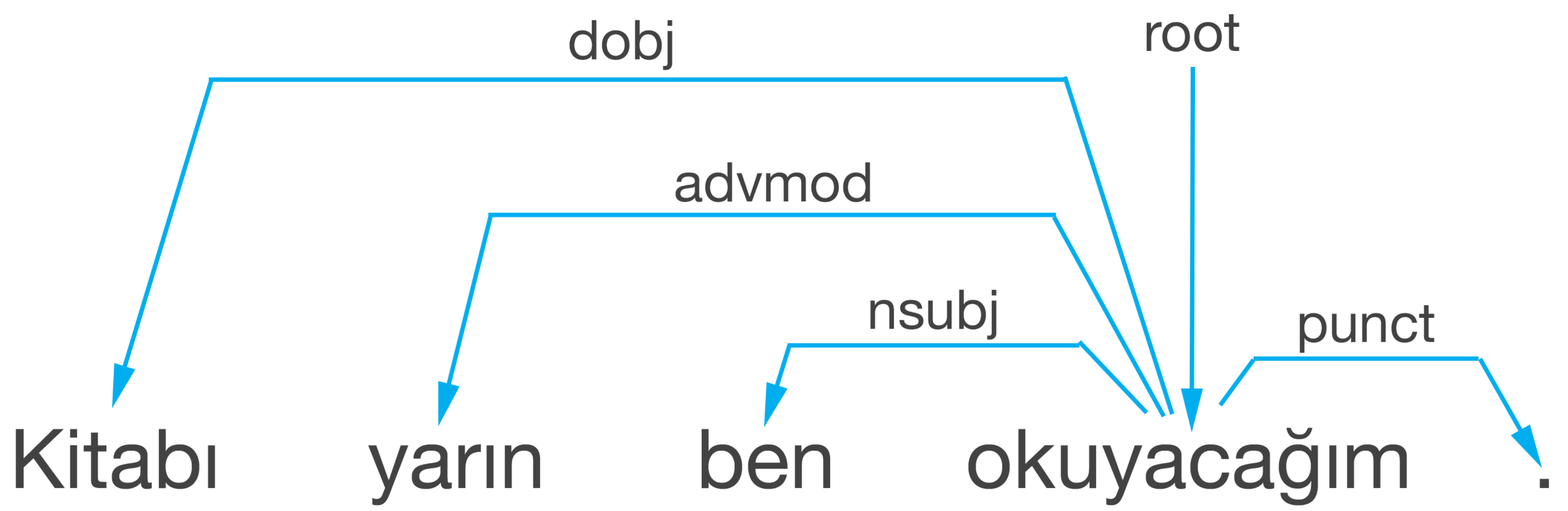
To derive this, we need tons of movement & operations. The end result will be extremely complex even if we stick to the very reduced & simple trees from Ling101.



Dependency vs. Constituency trees

“Kitabı yarın ben okuyacağım.”

Yet this is not a challenge for dependency trees:



Towards dependency trees: Route 1

Annotations

- Interfaces:
  - UD Annotatrix
    https://github.com/jonorthwash/ud-annotatrix
  - Arborator
    https://arborator.ilpga.fr/q.cgi
  - WebAnno
    https://webanno.github.io/webanno/
  - Conllu Editor
    https://github.com/Orange-OpenSource/conllueditor

Towards dependency trees: Route 2

Parsers

- Tools:
    - Stanza
      https://github.com/stanfordnlp/stanza
    - NLTK
      https://www.nltk.org
    - SpaCy
      https://github.com/explosion/spaCy

Towards dependency trees: Route 3

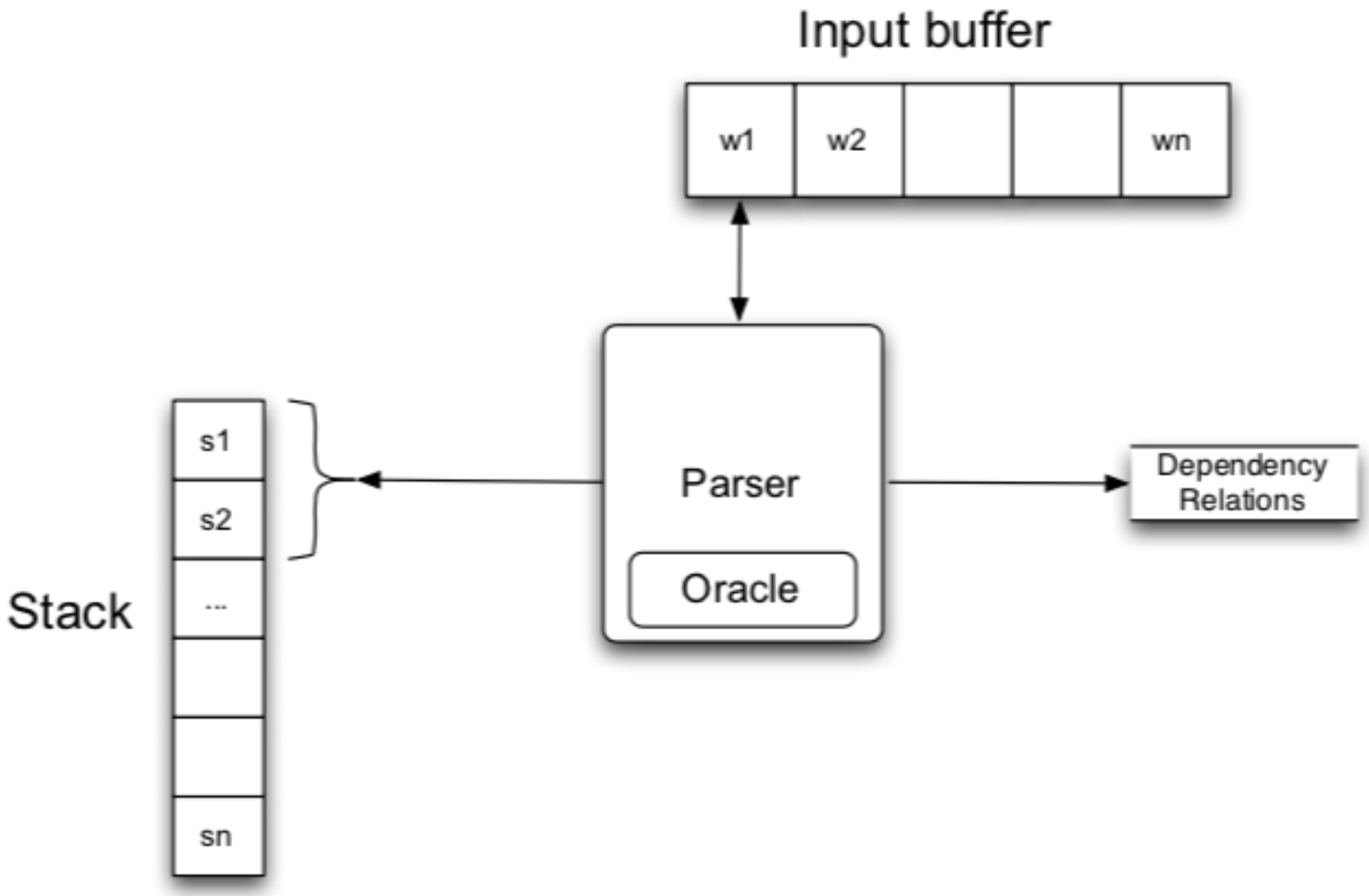
Making Our Own Parser

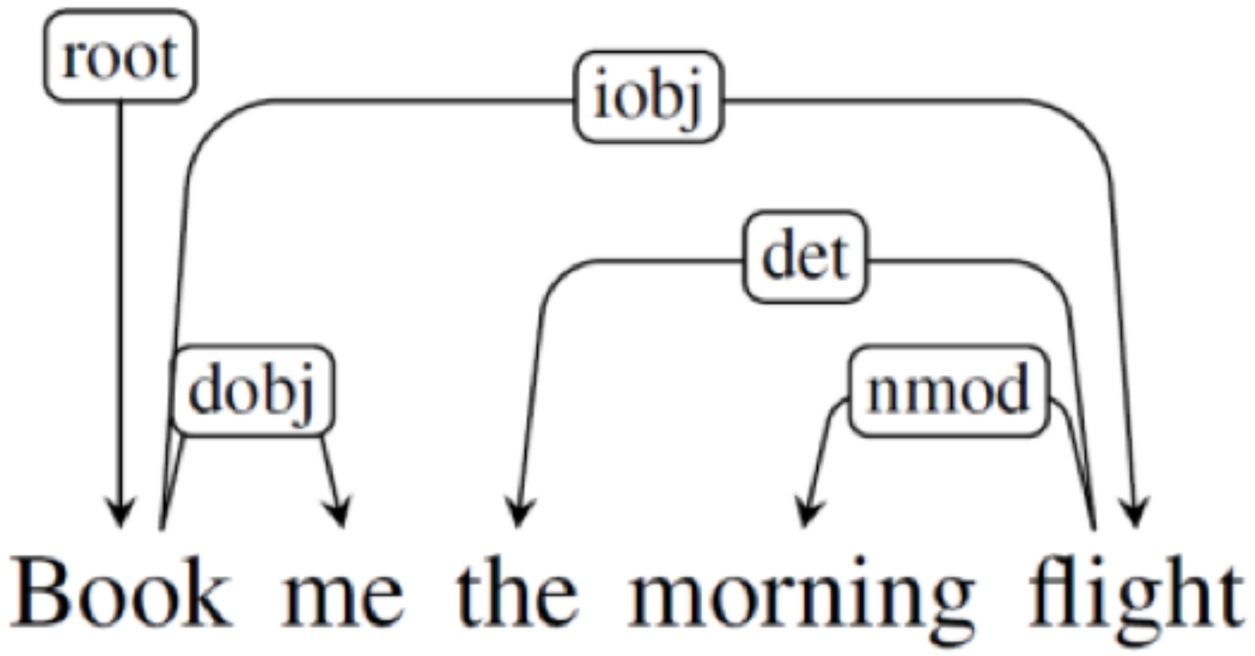
- Shift reduce parsing
- Graph-based parsing
- Maximum spanning tree
- ….

Towards dependency trees: Route 3

Shift reduce parsing

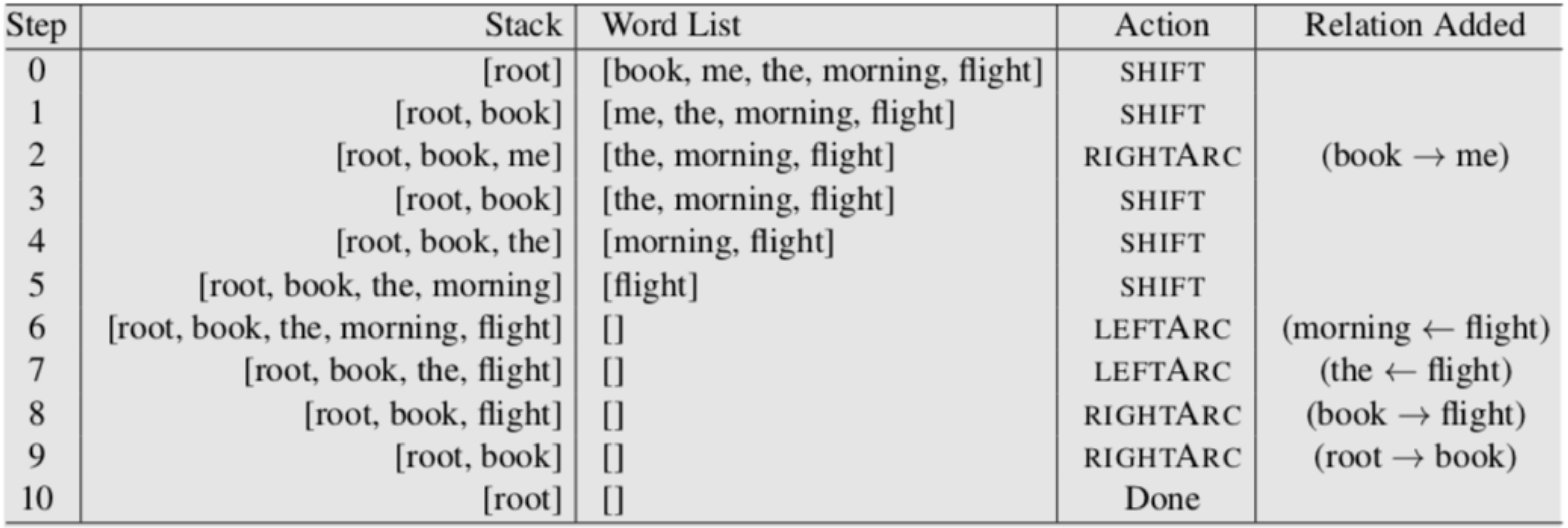
- The most straightforward one. Uses 3 transition operations.



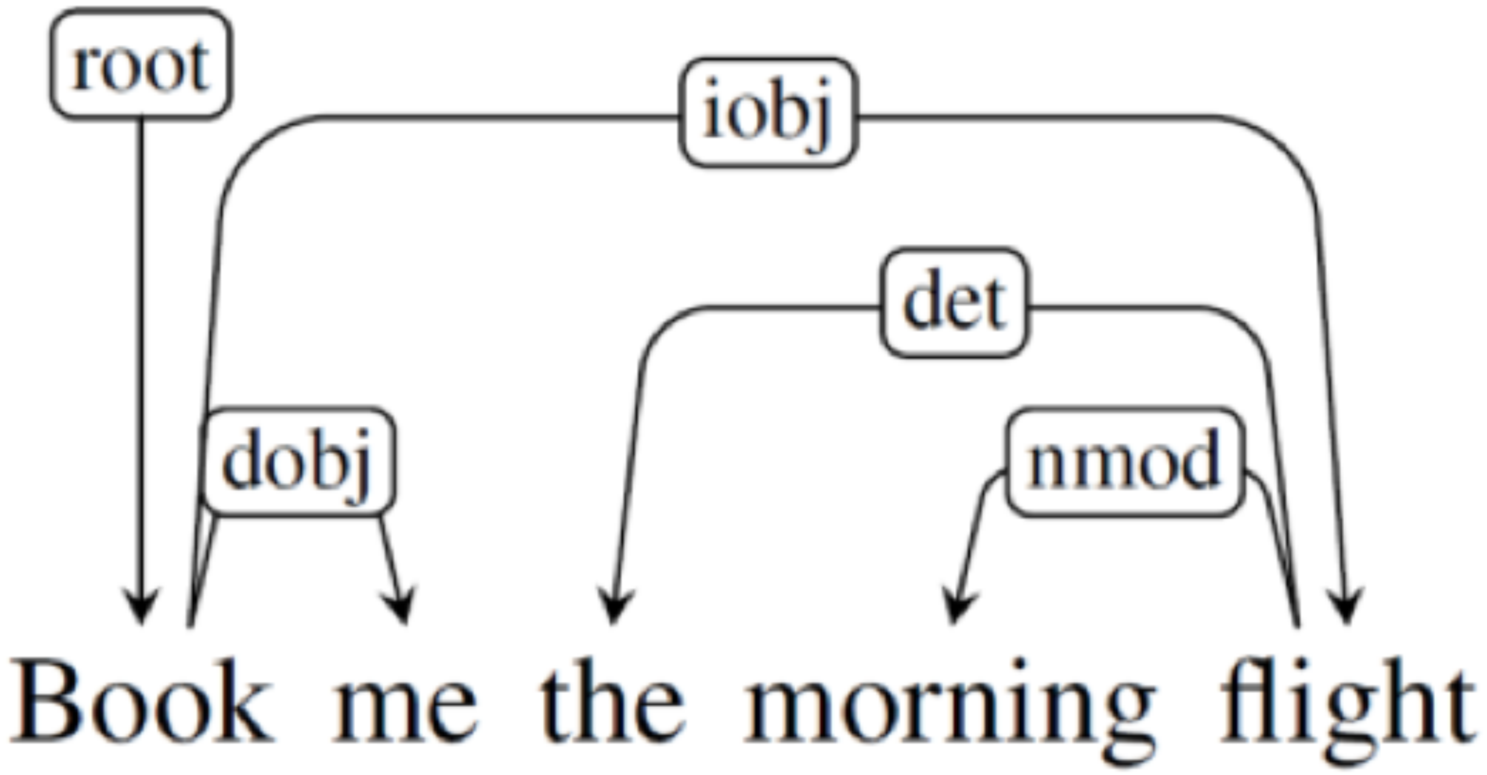


Towards dependency trees: Route 3

Shift reduce parsing



Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	



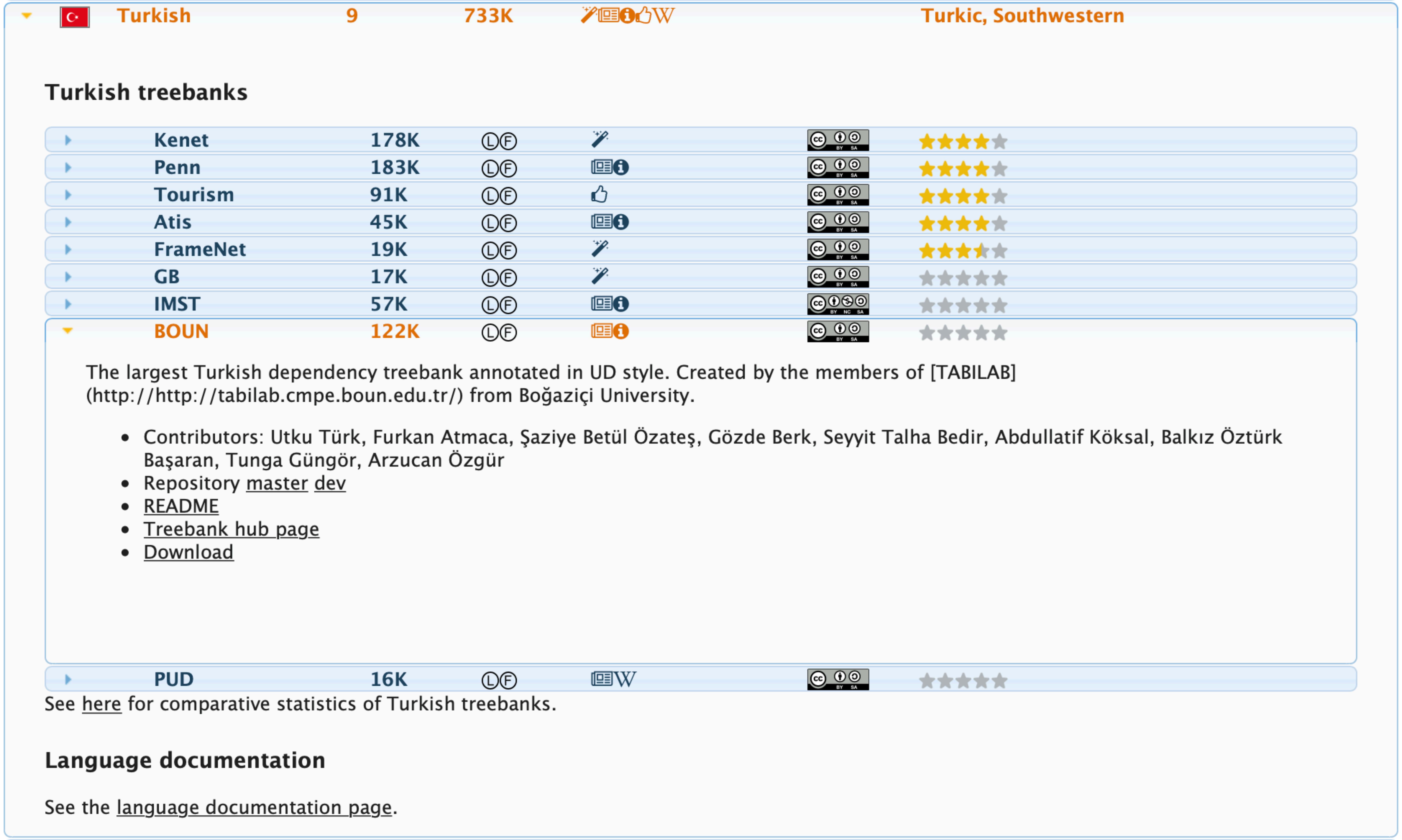
The Universal Dependencies framework

- There are various dependency grammar frameworks. Two most popular ones are Stanford Dependencies and Universal Dependencies (see references for related work).

- UD is **open source** and focuses on morpho-syntax. It also supports **multilingual corpora.**

- Currently the UD database is much larger: Over 100 languages, more than 200 treebanks.
    - Turkish has **12th largest database** in UD which encompasses more than 700,000 tokens!

The Universal Dependencies framework



Turkish	9	733K	🪄📰ℹ👍W	Turkic, Southwestern

Turkish treebanks

Kenet	178K	ⓁⒻ	🪄	cc BY SA	★★★★☆
Penn	183K	ⓁⒻ	📰ℹ	cc BY SA	★★★★☆
Tourism	91K	ⓁⒻ	👍	cc BY SA	★★★★☆
Atis	45K	ⓁⒻ	📰ℹ	cc BY SA	★★★★☆
FrameNet	19K	ⓁⒻ	🪄	cc BY SA	★★★½☆
GB	17K	ⓁⒻ	🪄	cc BY SA	☆☆☆☆☆
IMST	57K	ⓁⒻ	📰ℹ	cc BY NC SA	☆☆☆☆☆
BOUN	122K	ⓁⒻ	📰ℹ	cc BY SA	☆☆☆☆☆
PUD	16K	ⓁⒻ	📰W	cc BY SA	☆☆☆☆☆

BOUN:

The largest Turkish dependency treebank annotated in UD style. Created by the members of [TABILAB](http://http://tabilab.cmpe.boun.edu.tr/) from Boğaziçi University.

- Contributors: Utku Türk, Furkan Atmaca, Şaziye Betül Özateş, Gözde Berk, Seyyit Talha Bedir, Abdullatif Köksal, Balkız Öztürk Başaran, Tunga Güngör, Arzucan Özgür
- Repository master dev
- README
- Treebank hub page
- Download

See here for comparative statistics of Turkish treebanks.

Language documentation

See the language documentation page.

Use cases of UD style treebanks: NLP applications

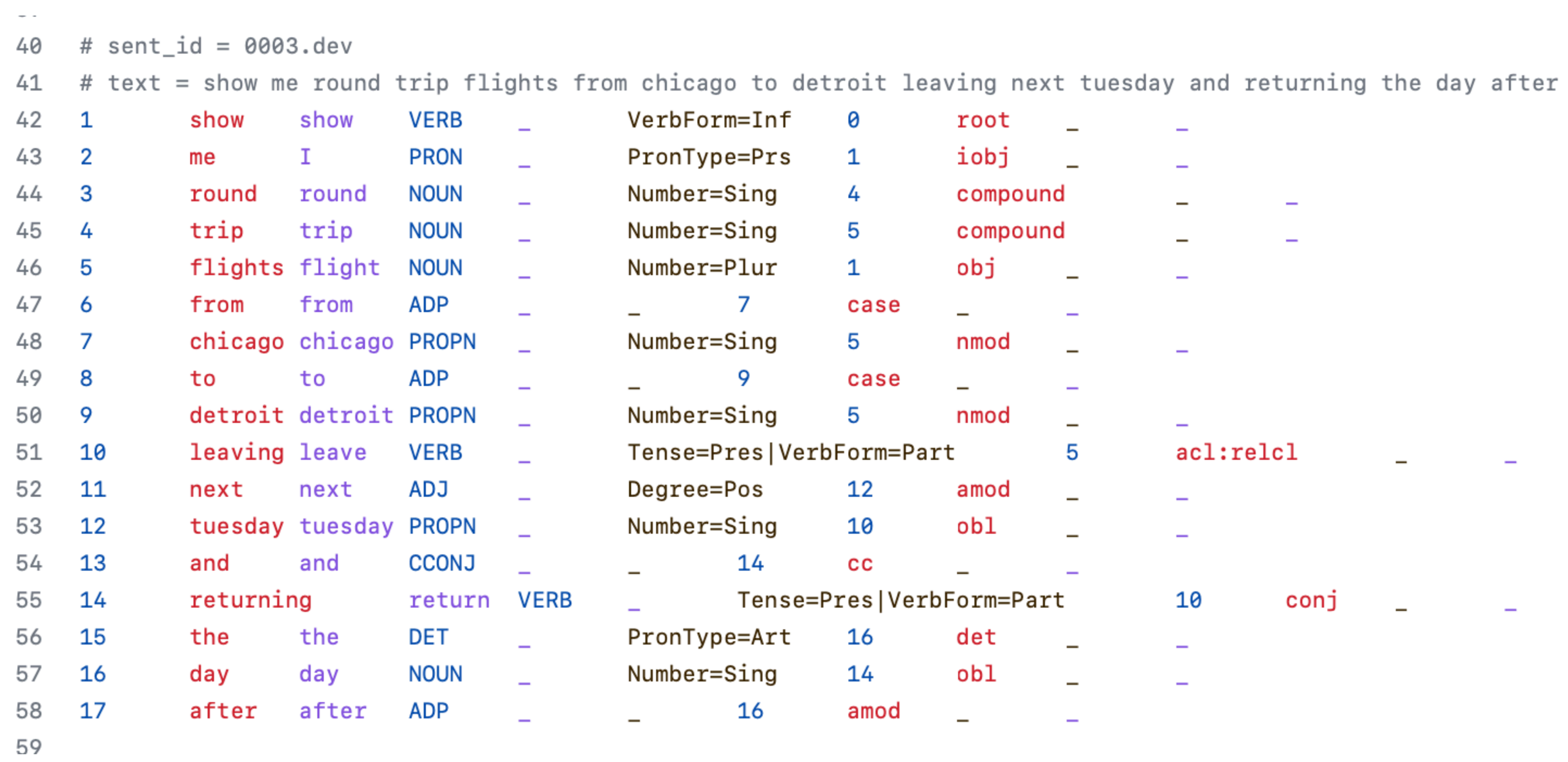
- Parsing is a building block for **downstream tasks,** thus a wide range of NLP applications use dependency parsing for many purposes including **disambiguation**:

  “Kötü bir ürün değil.” =/= “Kötü bir ürün.”
  “X değil Y olsa muhteşem.” =/= “Muhteşem.”

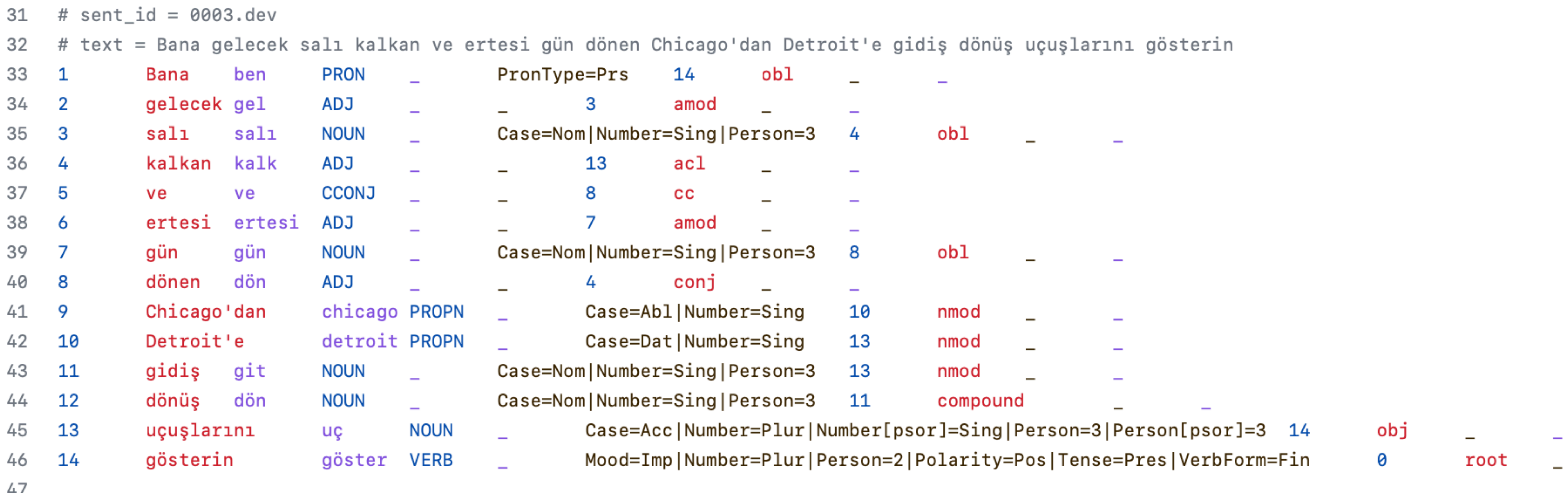
- Sentiment analysis, classification of customer feedback, named entity recognition (NER), question answering, dialogue systems…

Use cases of UD style treebanks: Theoretical linguistics

- UD style treebanks can be (and are) used in **theoretical and/or experimental linguistics** and **quantificational research** as well!
- It offers:
  - **Structured** data
  - Syntactic analysis
  - Coherent and **detailed representations**
  - Ability to do **cross-linguistic research**
  - Can be used for **data generation** (experimental items etc.)



```
# sent_id = 0003.dev
# text = show me round trip flights from chicago to detroit leaving next tuesday and returning the day after
1       show    show    VERB    _       VerbForm=Inf    0       root    _       _
2       me      I       PRON    _       PronType=Prs    1       iobj    _       _
3       round   round   NOUN    _       Number=Sing     4       compound        _       _
4       trip    trip    NOUN    _       Number=Sing     5       compound        _       _
5       flights flight  NOUN    _       Number=Plur     1       obj     _       _
6       from    from    ADP     _       _       7       case    _       _
7       chicago chicago PROPN   _       Number=Sing     5       nmod    _       _
8       to      to      ADP     _       _       9       case    _       _
9       detroit detroit PROPN   _       Number=Sing     5       nmod    _       _
10      leaving leave   VERB    _       Tense=Pres|VerbForm=Part        5       acl:relcl       _       _
11      next    next    ADJ     _       Degree=Pos      12      amod    _       _
12      tuesday tuesday PROPN   _       Number=Sing     10      obl     _       _
13      and     and     CCONJ   _       _       14      cc      _       _
14      returning       return  VERB    _       Tense=Pres|VerbForm=Part        10      conj    _       _
15      the     the     DET     _       PronType=Art    16      det     _       _
16      day     day     NOUN    _       Number=Sing     14      obl     _       _
17      after   after   ADP     _       _       16      amod    _       _
```

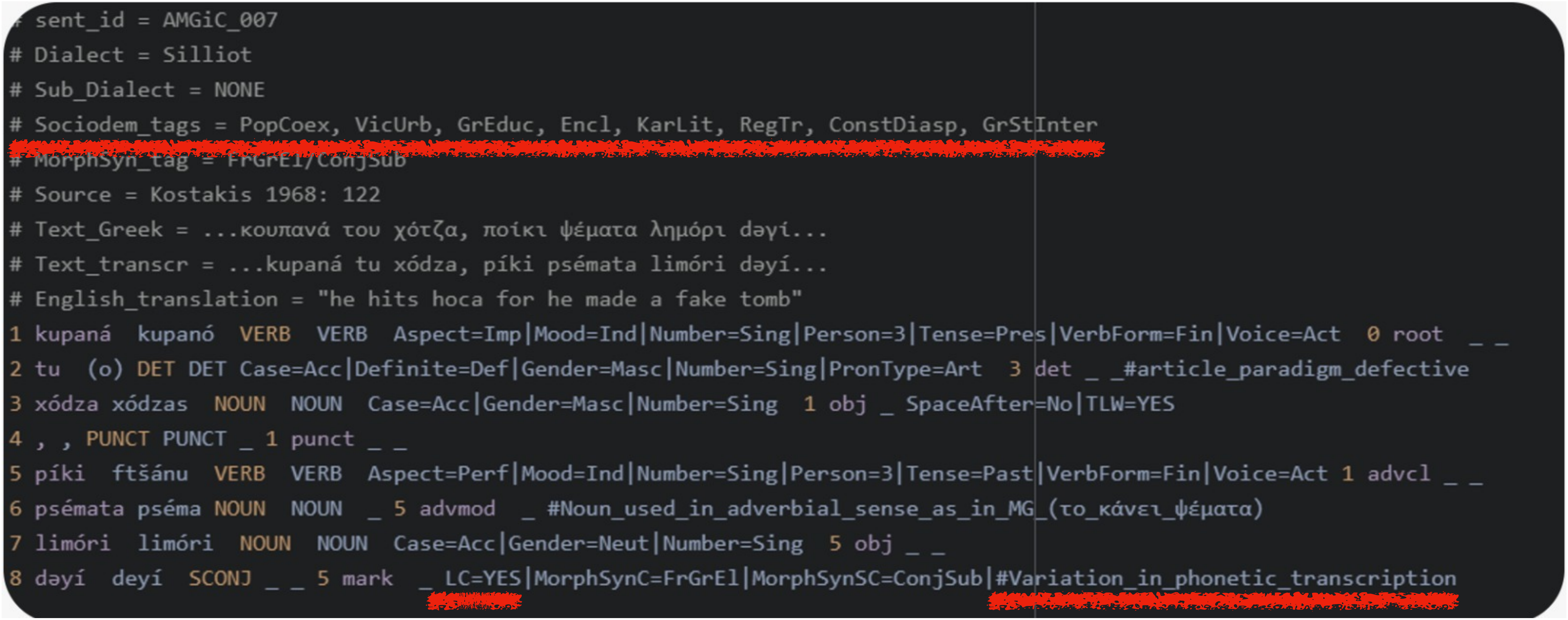


```
# sent_id = 0003.dev
# text = Bana gelecek salı kalkan ve ertesi gün dönen Chicago'dan Detroit'e gidiş dönüş uçuşlarını gösterin
1       Bana    ben     PRON    _       PronType=Prs    14      obl     _       _
2       gelecek gel     ADJ     _       _       3       amod    _       _
3       salı    salı    NOUN    _       Case=Nom|Number=Sing|Person=3   4       obl     _       _
4       kalkan  kalk    ADJ     _       _       13      acl     _       _
5       ve      ve      CCONJ   _       _       8       cc      _       _
6       ertesi  ertesi  ADJ     _       _       7       amod    _       _
7       gün     gün     NOUN    _       Case=Nom|Number=Sing|Person=3   8       obl     _       _
8       dönen   dön     ADJ     _       _       4       conj    _       _
9       Chicago'dan     chicago PROPN   _       Case=Abl|Number=Sing    10      nmod    _       _
10      Detroit'e       detroit PROPN   _       Case=Dat|Number=Sing    13      nmod    _       _
11      gidiş   git     NOUN    _       Case=Nom|Number=Sing|Person=3   13      nmod    _       _
12      dönüş   dön     NOUN    _       Case=Nom|Number=Sing|Person=3   11      compound        _       _
13      uçuşlarını      uç      NOUN    _       Case=Acc|Number=Plur|Number[psor]=Sing|Person=3|Person[psor]=3  14      obj     _       _
14      gösterin        göster  VERB    _       Mood=Imp|Number=Plur|Person=2|Polarity=Pos|Tense=Pres|VerbForm=Fin      0       root    _
```

(ATIS data from the UD database)

Use cases of UD style treebanks: Preserving endangered languages

- Language documentation is one of the biggest concerns regarding endangered languages. UD style treebanks allow this in a sophisticated and functional way:
  - **Annotated** data (syntactic, morphological, extra linguistic…)
  - **Accessible**: open-source
  - Can be read by computers
  - Allows quantificational work
  - Treebanks can be updated, re-annotated, extended



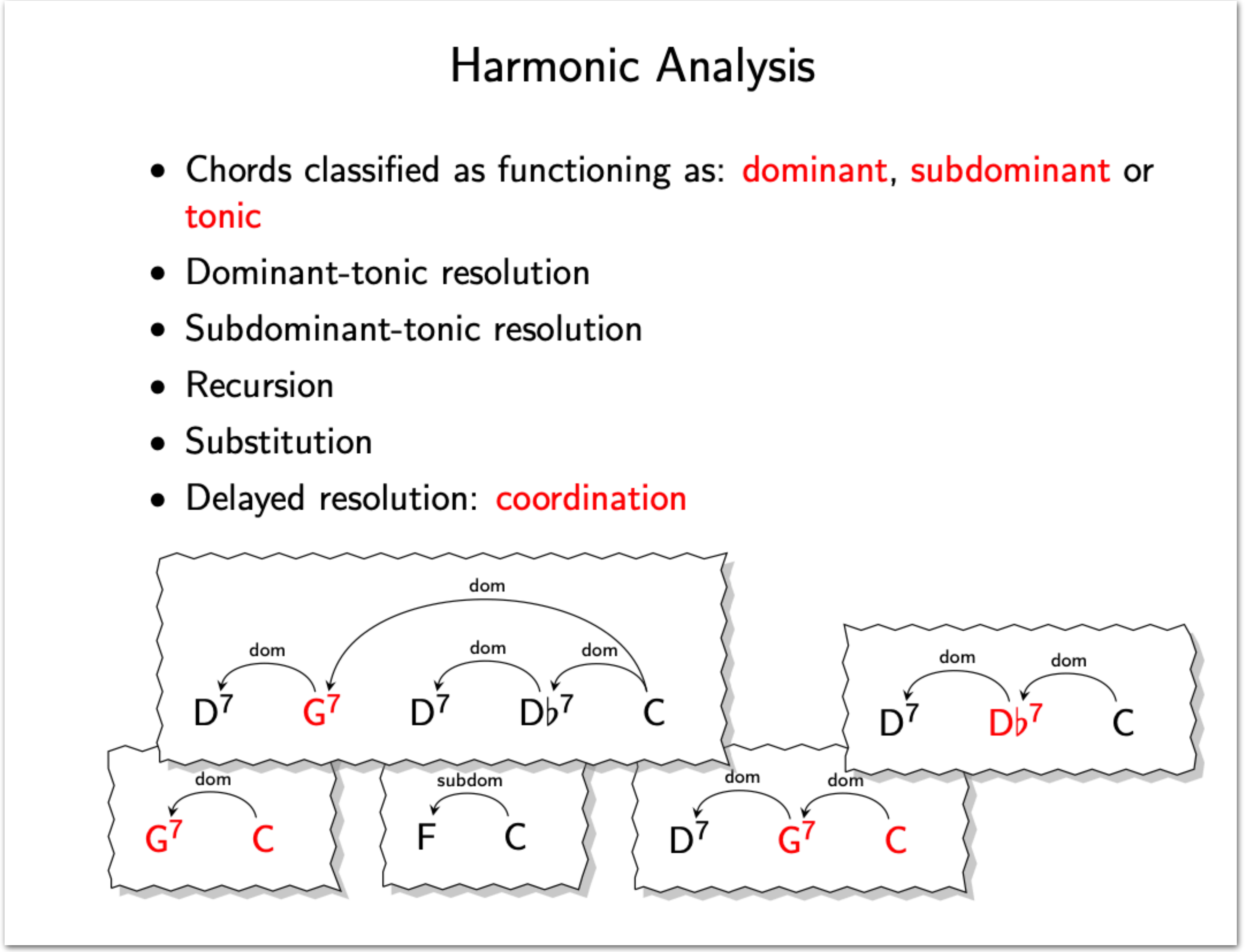
```
# sent_id = AMGiC_007
# Dialect = Silliot
# Sub_Dialect = NONE
# Sociodem_tags = PopCoex, VicUrb, GrEduc, Encl, KarLit, RegTr, ConstDiasp, GrStInter
# MorphSyn_tag = FrGrEl/ConjSub
# Source = Kostakis 1968: 122
# Text_Greek = ...κουπανά του χότζα, ποίκι ψέματα λημόρι dəγí...
# Text_transcr = ...kupaná tu xódza, píki psémata limóri dəyí...
# English_translation = "he hits hoca for he made a fake tomb"
1 kupaná  kupanó  VERB  VERB  Aspect=Imp|Mood=Ind|Number=Sing|Person=3|Tense=Pres|VerbForm=Fin|Voice=Act  0 root  _ _
2 tu  (o) DET DET Case=Acc|Definite=Def|Gender=Masc|Number=Sing|PronType=Art  3 det _ _#article_paradigm_defective
3 xódza xódzas  NOUN  NOUN  Case=Acc|Gender=Masc|Number=Sing  1 obj _ SpaceAfter=No|TLW=YES
4 , , PUNCT PUNCT _ 1 punct _ _
5 píki  ftšánu  VERB  VERB  Aspect=Perf|Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin|Voice=Act 1 advcl _ _
6 psémata pséma NOUN  NOUN  _ 5 advmod  _ #Noun_used_in_adverbial_sense_as_in_MG_(το_κάνει_ψέματα)
7 limóri  limóri  NOUN  NOUN  Case=Acc|Gender=Neut|Number=Sing  5 obj _ _
8 dəyí  deyí  SCONJ _ _ 5 mark  _ LC=YES|MorphSynC=FrGrEl|MorphSynSC=ConjSub|#Variation_in_phonetic_transcription
```

How to contribute

- We have a **constant** need of more contributors! You can:
  - Join UD mailing list*
  - Do annotations
  - Provide data
  - Find issues

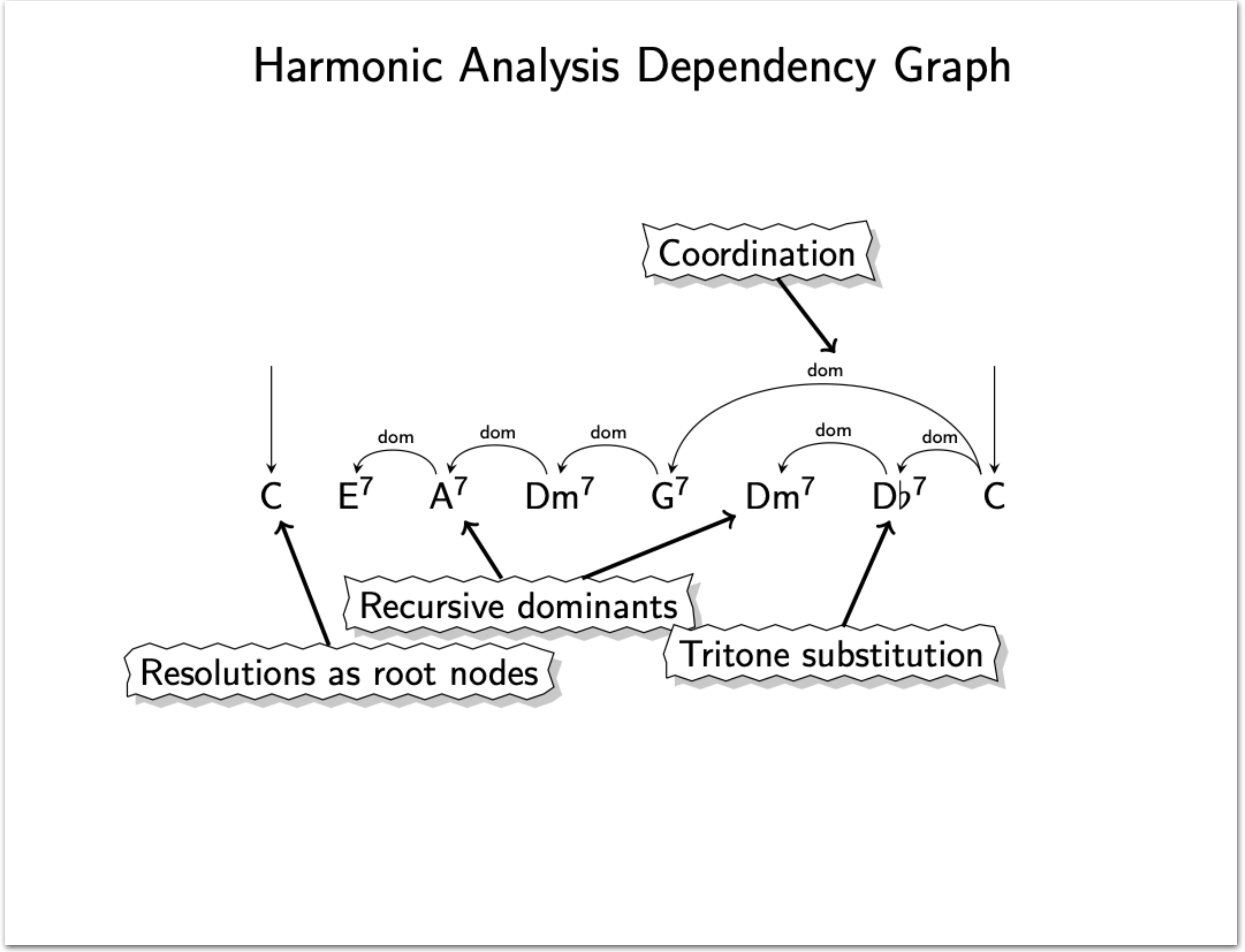
*https://cl.lingfil.uu.se/mailman/listinfo/ud

Trivia: Dependency grammar can also be used in music theory!



Harmonic Analysis

- Chords classified as functioning as: dominant, subdominant or tonic
- Dominant-tonic resolution
- Subdominant-tonic resolution
- Recursion
- Substitution
- Delayed resolution: coordination



Harmonic Analysis Dependency Graph

Coordination

Recursive dominants

Resolutions as root nodes

Tritone substitution

http://mark.granroth-wilding.co.uk/files/mml2012_article.pdf

Dependency Treebanks

- UD Treebank
  https://github.com/UniversalDependencies

- UD Turkish BOUN
  https://github.com/boun-tabi/UD_Turkish-BOUN

- UD Penn Turkish
  https://github.com/UniversalDependencies/UD_Turkish-Penn

Queries

- You can query UD treebanks online:
  - **SETS treebank search** maintained by the University of Turku
  - **PML Tree Query** maintained by the Charles University in Prague
  - **Kontext** maintained by the Charles University in Prague
  - **Grew-match** maintained by Inria in Nancy
  - **INESS** maintained by the University of Bergen

Resources

- More on DP and conversions b/w constituency & dependency: https://web.stanford.edu/~jurafsky/slp3/14.pdf
- Detailed explanations of dependency tags: https://universaldependencies.org/u/dep/
- Stanford Dependencies: https://nlp.stanford.edu/software/stanford-dependencies.shtml
- Universal Dependencies: https://universaldependencies.org
- CoNNL format guide: https://universaldependencies.org/docs/format.html
- UD Tools https://universaldependencies.org/tools.html#arborator

Literature

- Literature on UD:
  https://direct.mit.edu/coli/article/47/2/255/98516/Universal-Dependencies
  https://arxiv.org/abs/2004.10643
  https://universaldependencies.org/introduction.html

- Literature on Stanford Dependencies:
  http://www.lrec-conf.org/proceedings/lrec2014/pdf/1062_Paper.pdf
  https://aclanthology.org/W13-3721.pdf

- Endangered languages & Dependency Treebanks:
  https://aclanthology.org/2020.udw-1.21/
  https://aclanthology.org/2021.tlt-1.8.pdf

Contact Info & Materials

- Büşra Marşan
  busra.marsan@boun.edu.tr

- Scan the QR code to access the materials I used in this workshop —including this presentation.

SCAN ME