Ling/CSE 472:
Introduction to Computational Linguistics

5/7: Dependency parsing

Overview

- Grammatical dependencies
- Dependency grammar
- Dependency treebanks
- Dependency parsing
- Reading questions (with headers)
- Questions about milestone 2

Grammatical Dependencies

- Relate words in the sentence to each other
- A labeled with the type of dependency
- Are typically represented as graphs (sometimes trees)
  - Where each node is a word in the sentence
  - Where word in the sentence is (usually) a node

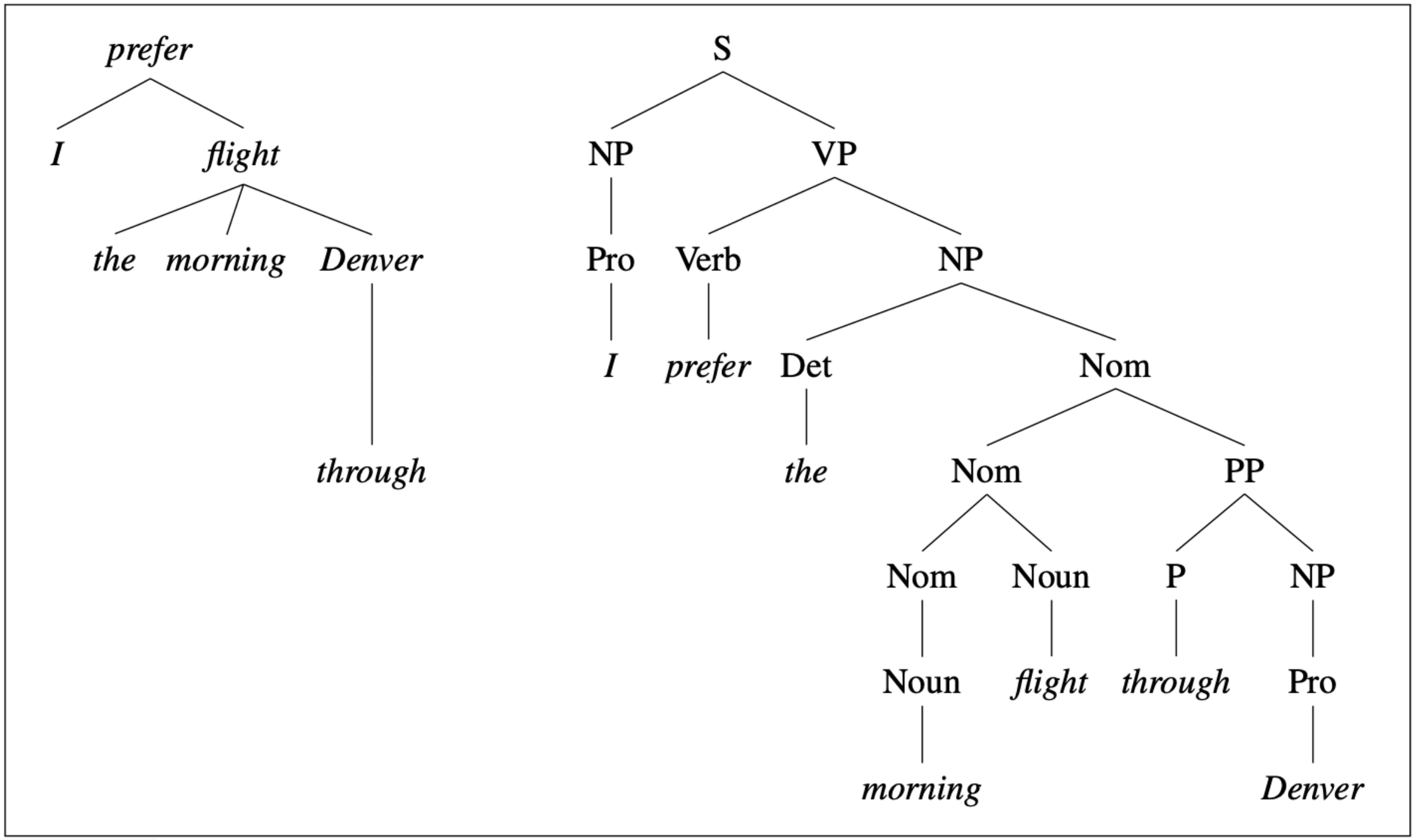
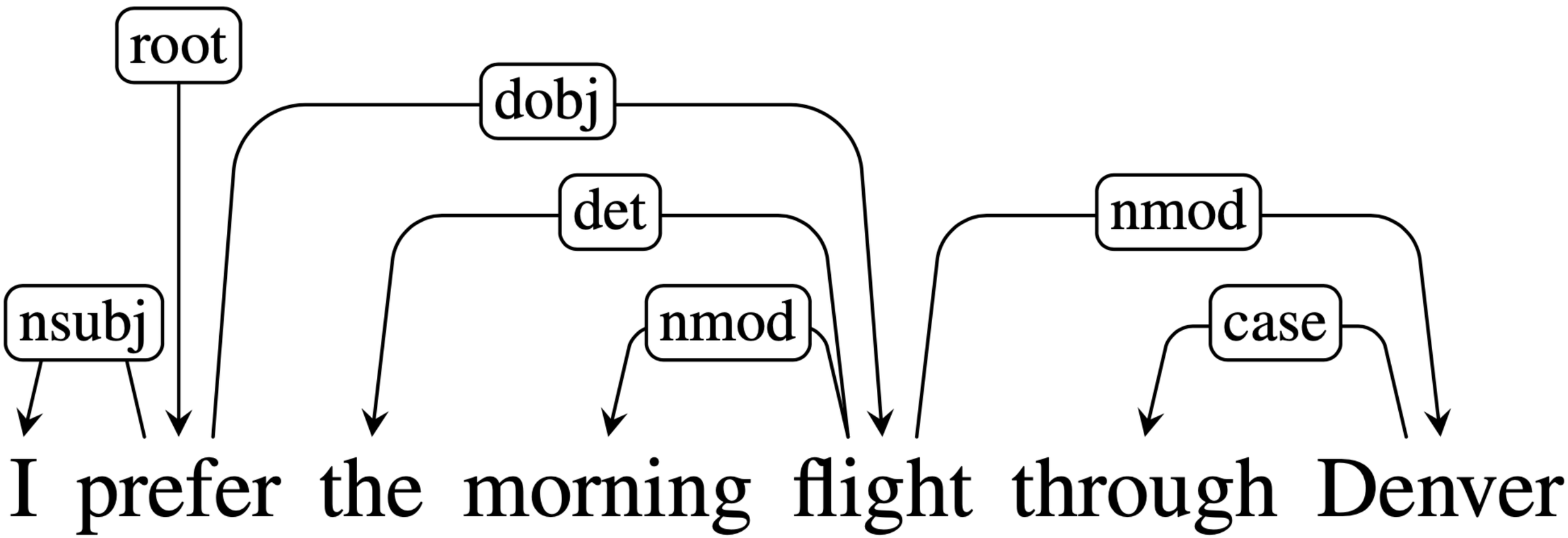


Figure 15.1 A dependency-style parse alongside the corresponding constituent-based analysis for *I prefer the morning flight through Denver.*


root
dobj
det
nmod
nsubj
nmod
case
I prefer the morning flight through Denver

Dependency Grammar

- Theoretical foundations: Tesnière 1959, Mel’čuk 1988, Hudson 1984, Sgall et al 1986

- Focus not on grammaticality (“What’s a possible sentence?”) but on grammatical structure, given a string

Dependency Treebanks: Universal Dependencies

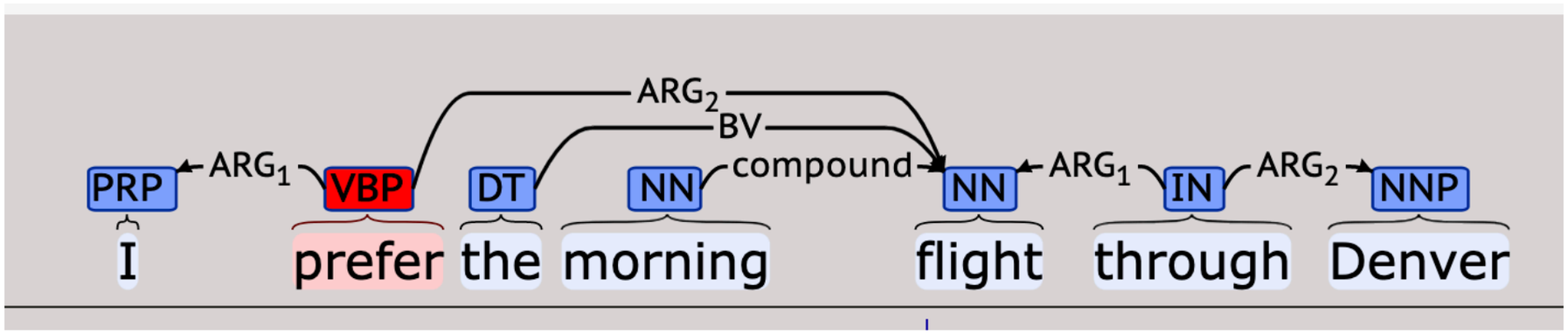
- https://universaldependencies.org/
- Builds on:
  - Stanford dependencies (LFG-inspired transformation of CFG representations for English from the Stanford parser)
  - Theoretical work on dependency grammar
  - “Universal” POS tagset developed initially for cross-linguistic error analysis (McDonald and Nivre 2007)

What is needed for UD to be successful? (from universaldependencies.org/introduction.html)

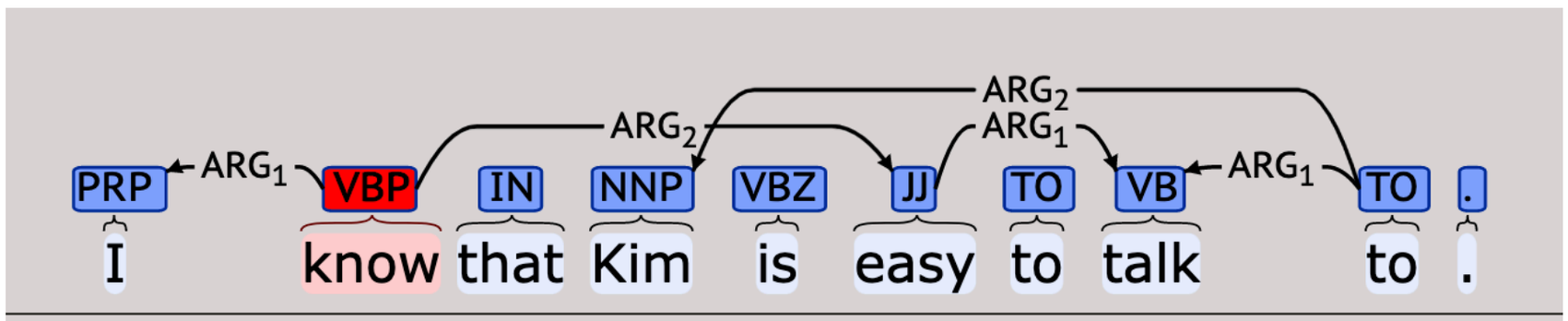
- The secret to understanding the design and current success of UD is to realize that the design is a very subtle compromise between approximately 6 things:
  - UD needs to be satisfactory on linguistic analysis grounds for individual languages.
  - UD needs to be good for linguistic typology, i.e., providing a suitable basis for bringing out cross-linguistic parallelism across languages and language families.
  - UD must be suitable for rapid, consistent annotation by a human annotator.
  - UD must be suitable for computer parsing with high accuracy.
  - UD must be easily comprehended and used by a non-linguist, whether a language learner or an engineer with prosaic needs for language processing. We refer to this as seeking a habitable design, and it leads us to favor traditional grammar notions and terminology.
  - 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.

Dependency Treebanks outside UD

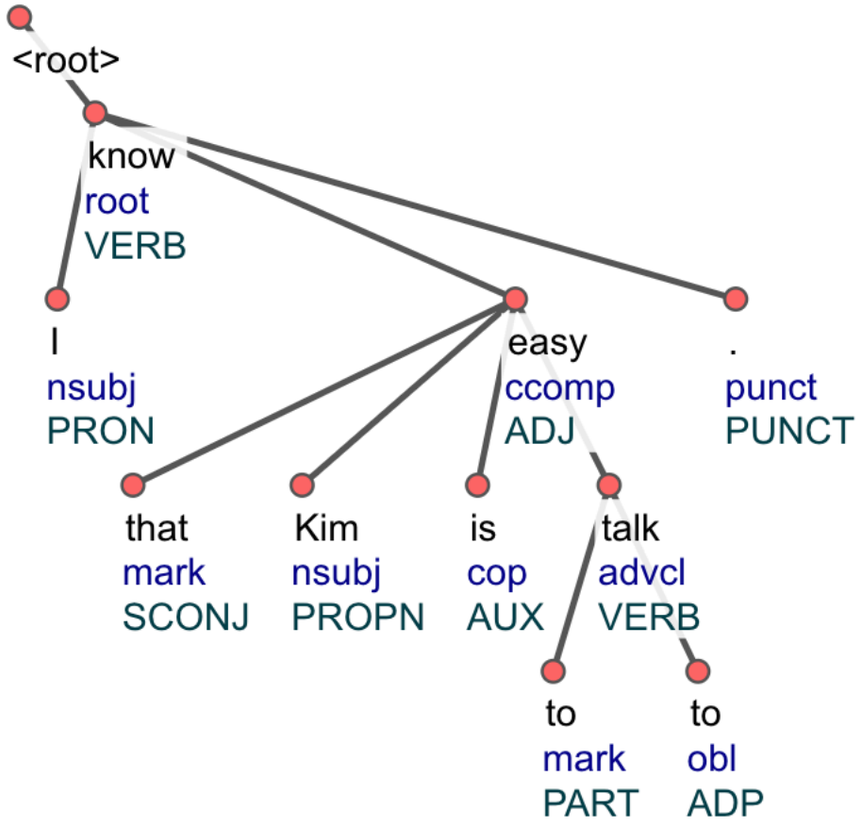
- Richer grammatical formalisms such as HPSG can be ‘boiled down’ to dependency representations
  - Syntactic OR semantic dependencies (Ivanova et al 2012)



DM v. UD



I know that Kim is easy to talk to .



Dependency Parsing

- Transition-based v. graph-based
- Feature templates v. neural
- Source of training data

Dependency Parsing

function DEPENDENCYPARSE(*words*) **returns** dependency tree

state ← {[root], [*words*], [] } ; initial configuration
while *state* **not final**
t ← ORACLE(*state*) ; choose a transition operator to apply
state ← APPLY(*t*, *state*) ; apply it, creating a new state
return *state*

Figure 15.6 A generic transition-based dependency parser

Dependency 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	

Figure 15.7 Trace of a transition-based parse.

Reading questions: Why?

- This is a very broad question, but what exactly is the goal with these dependency structures? We went over an algorithm in class, and these chapters went over more relations. After these are made, and possibly hand corrected, what does the computer do with these? Is having this structure just an important thing in general to have when processing language?

Reading questions: Root

- Is the “root” always the main verb? Or are there times when the root is something else?

Reading questions:
Constituency v. dependency parsing

- In the translation process from constituency structure to dependency structure, what are head-finding rules?

- What are the benefits of constituency parsing over dependency parsing? Is it just that dependency parsing isn't always necessary?

- Are there any advantages to choosing constituent based parsing over dependency parsing for a language with relatively free word order? It seems like it would be very hard to get a constituent based parser right with such a language, as there would be many productions for every type of phrase.

Reading questions: Constituency v. dependency parsing

- For languages that aren't as morphologically rich, would using this dependency parsing have no obvious net change in performance and convenience? (Compared to the other forms of parsing we've studied) Because the main idea, I'm getting is that it's most beneficial to languages that are morphologically rich.

- The chapter notes that without a tool like dependency grammar it can be very difficult to deal with free word order and morphological richness. Free word order seems pretty happily managed by the basic premise of dependency parsing (at least in theory), but it’s not clear to me what degree of morphological complexity demands strategies like making the vertices represent stems and affixes instead of words - like, would that be necessary in English, or is it okay to just have different tags for, e.g., said and says? Is that the kind of decision you might make based on the type of annotation you have in the first place?

Reading questions:
Constituency v. dependency parsing

- Coming from the CCG topic, which was one of the topic for this week's blog posts, the idea of dependency graph seemed similar to the functions in CCG in the way that the constraint that allows only one incoming arc to be presented in each vertex except for the root vertex as functions in CCG are required to be a single-argument function and so on. Are all of these formalisms differ from each other such that they are fundamentally not equivalent to each other, or they are different views encoding the same information?

Reading questions:
Morphologically complex languages

- I curious about how dependency phrasing deals with language that heavily uses affixes for grammar (agglutination). For example, "from your house" in Turkish is "ev-ler-iniz-den", where CASE and NMOD are in one word.

Reading questions: Annotation

- Section 15.3 mentioned that dependency treebanks have been manually annotated for morphologically rich languages (and possibly languages with non-nominative-accusative morphosyntactic alignment like ergative-absolute and tripartite languages). Would manually annotating these treebanks not be an ambiguous task between different annotators? What developments have been made to reduce ambiguity and bias between annotating these treebanks?

- Is it possible for there to be multiple different possible dependency relations between two words? If so, how is the "correct" one determined?

Reading questions: Annotation

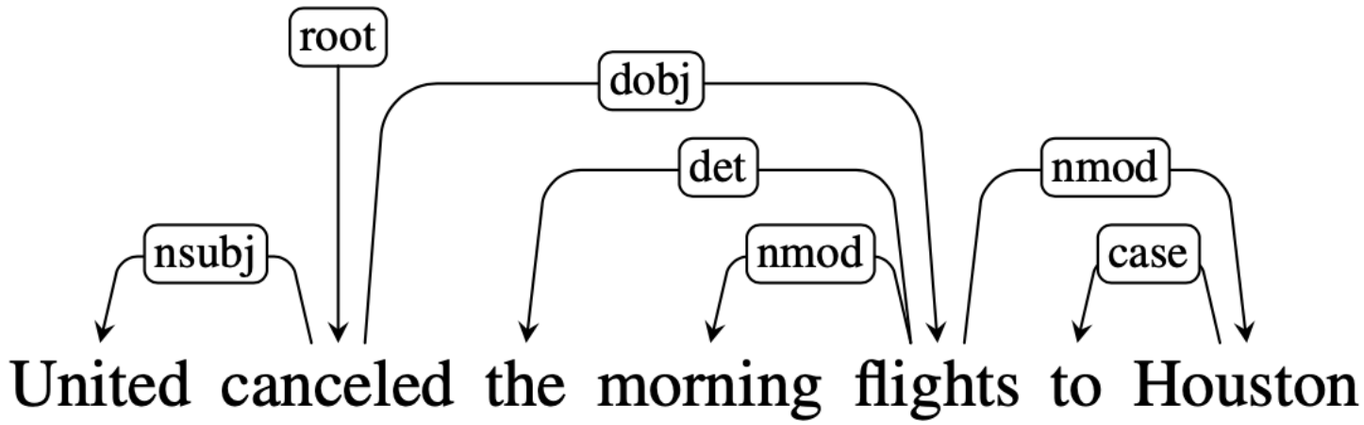
- In JM Ch 15.3, it mentions that the dependency treebanks are limited by the information in the original constituent trees, and this makes such treebanks to be directly developed by human annotators. Would it possible to train supervised ML models that address each of the issues mentioned in the text separately and use such models to create these dependency treebanks?

- Also, how is ambiguity dealt with in the generation of the dependency trees from the constituency trees? Because some sentences will have multiple constituency trees - will all of the options be translated to dependency trees? Will some have higher priority/weight?

Reading questions: Projectivity

- They say that this dependency tree (picture) is projective, but then I'm not sure what they mean by a "path from the head to every word that lies between the head and the dependent", because cancelled isn't connected with the or morning and flights isn't connected with to. Even though those words may modify another head, they're still words in between? (15.2)

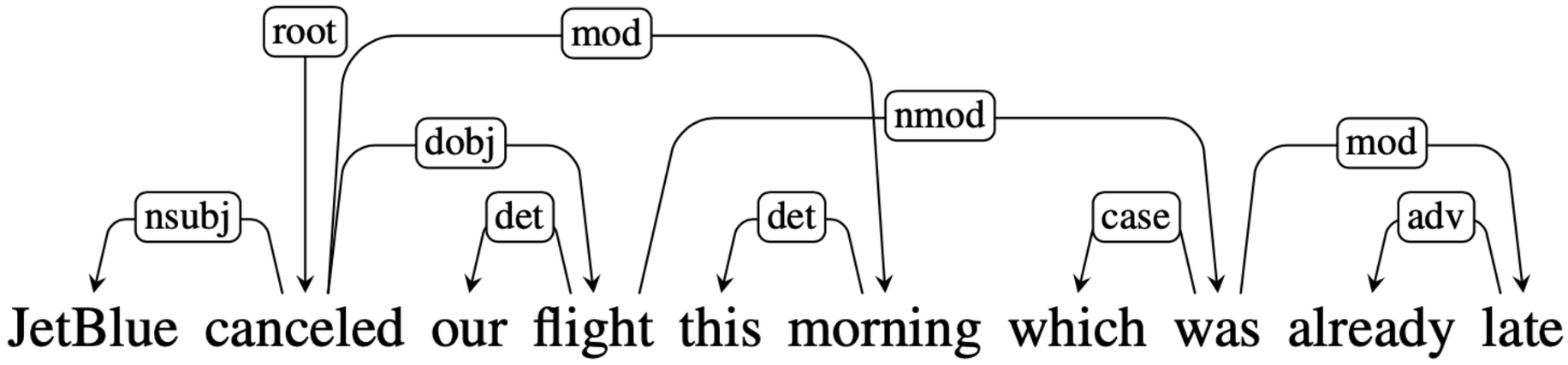
(15.2)



Reading questions: Projectivity

- They say that this dependency tree (picture) is projective, but then I'm not sure what they mean by a "path from the head to every word that lies between the head and the dependent", because cancelled isn't connected with the or morning and flights isn't connected with to. Even though those words may modify another head, they're still words in between? (15.2)

(15.3)



Reading questions: Projectivity

- Because treebanks are generated from context-free grammars and thus are all projective, is there a lack of non-projective data? If so, how can that data bank be enlarged so as to provide better parsing of morphologically rich languages and languages with free word order?

- Are there efforts to expand treebanks to include non-projective trees, such as for treebanks that were built from existing constituent structure treebanks, or is non-projectivity a relatively infrequent issue? Mostly considering English here.

Reading questions: Projectivitys

- Is it more important that we are just aware of the idea of projectivity, so that when we use a certain parsing algorithm and we know that it can only handle projective sentences, we can deal with that accordingly? Or is there another particular advantage to using projective vs. non-projective sentences when testing/training models?

- How relative is projectivity to clauses versus adjuncts - can non-projective sentences contain crossing due to either?

- How exactly is projectivity related to "the context-free nature of human languages"?

Reading questions: Other

- Would transition based dependency parsing support a language (I don't know if any exist) that have dependencies between words found far apart in the sentence?

- Are human annotations always the ground truth for determining the accuracy of dependency parsing models?

Milestone 2, due 5/15:
Complete project plan, 1st draft (2-3 pages)

- Submit a clear and detailed description of the package that you chose. (What is the tool for? How is it implemented (high level)? How is it evaluated? Why did you choose it?)

- Submit a clear and detailed description of the dataset that comes with the package, or, in exceptional circumstances, that you propose that you use for the project. How big is it? What is the format? Will you need to do any preprocessing? Etc

- Include URL for the package download site

- Include URL for the dataset download site

Milestone 2, due 5/15:
Complete project plan, 1st draft (2-3 pages)

- Copy the main results table from the paper and explain what sort of evaluation they use (what metrics etc.). Explain how the reader should interpret the table. What do these number mean with respect to what the tool is doing?

- What error analysis have the authors already done, if any?

- Include a clear plan of how you will perform your error analysis, with several examples:

- For instance, if you are doing error analysis for a parser: find several sentences which the parser does not parse, and explain how you would go about categorizing these errors and what kind of discussion you might develop about them.