6.1. Dependency Parsing

Edwin Simpson

Department of Computer Science,

University of Bristol, UK.

Words and Entities

- Previously, we used sequence labelling to label named entity spans
- Recognising entities means processing the **meaning** of a piece of text, e.g.:

  "United Airlines said Friday it has increased fares by $6."
- Which view of meaning do we use for NER?

Relational: the relationships between words encode meaning

Compositional: meaning is formed by combining units of text

Contextual: understanding a word from its context

Words and Entities

- Previously, we used sequence labelling to label named entity spans
- Recognising entities means processing the **meaning** of a piece of text, e.g.:

  "United Airlines said Friday it has increased fares by $6."
- HMM and CRF rely on the context of a word – its neighbours in a sentence

Relational: the relationships between words encode meaning

Compositional: meaning is formed by combining units of text

Contextual: understanding a word from its context

Words → Syntax

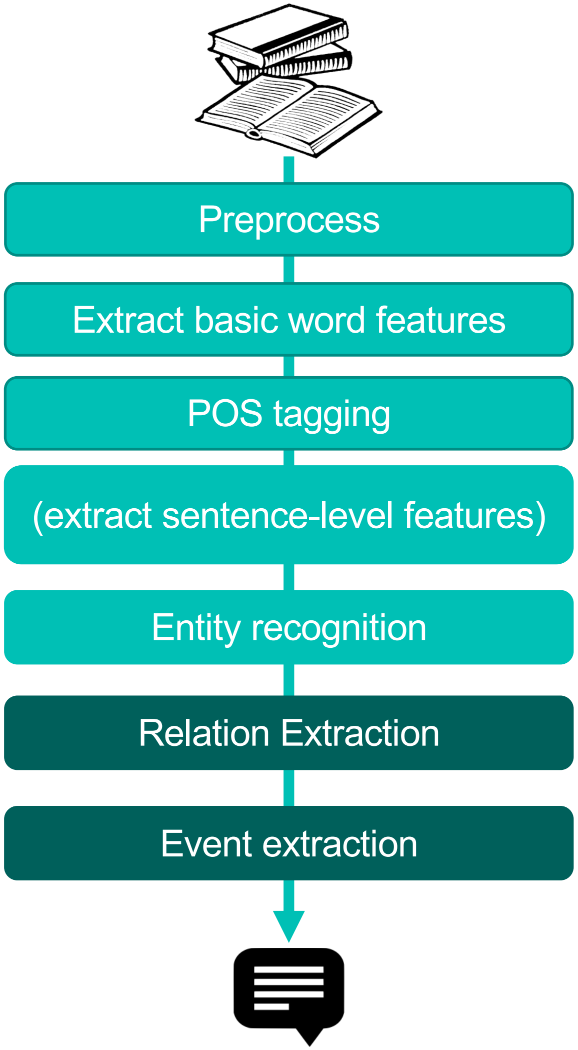
“United Airlines said Friday it has increased fares by $6.”

- There is a lot more information in here that we can only extract by analysing the **composition** of the whole sentence.
- The arrangement of words to form phrases and sentences is known as **syntax**
- We need an understanding of syntax to understand the full **meaning** of the sentence.

Words → Syntax → Meaning

"United Airlines said Friday it has increased fares by $6."

- Who increased fares?
- How much did they increase by?
- To answer this, we need to recognise how "United Airlines" and "$6" relate to the verb "increased"



Preprocess

Extract basic word features

POS tagging

(extract sentence-level features)

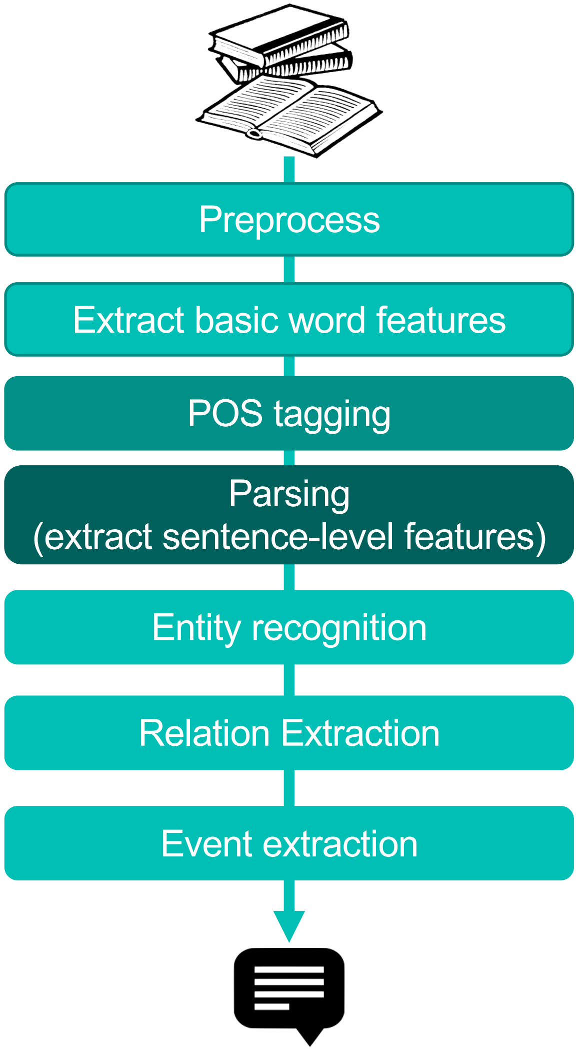
Entity recognition

Relation Extraction

Event extraction

Parsing

- Analysing the syntax of a sentence is known as **parsing**
- Parsing requires part of speech tags
- Parts of speech are categories of words that can be used in a particular way to construct phrases and sentences



Preprocess

Extract basic word features

POS tagging

Parsing
(extract sentence-level features)

Entity recognition

Relation Extraction

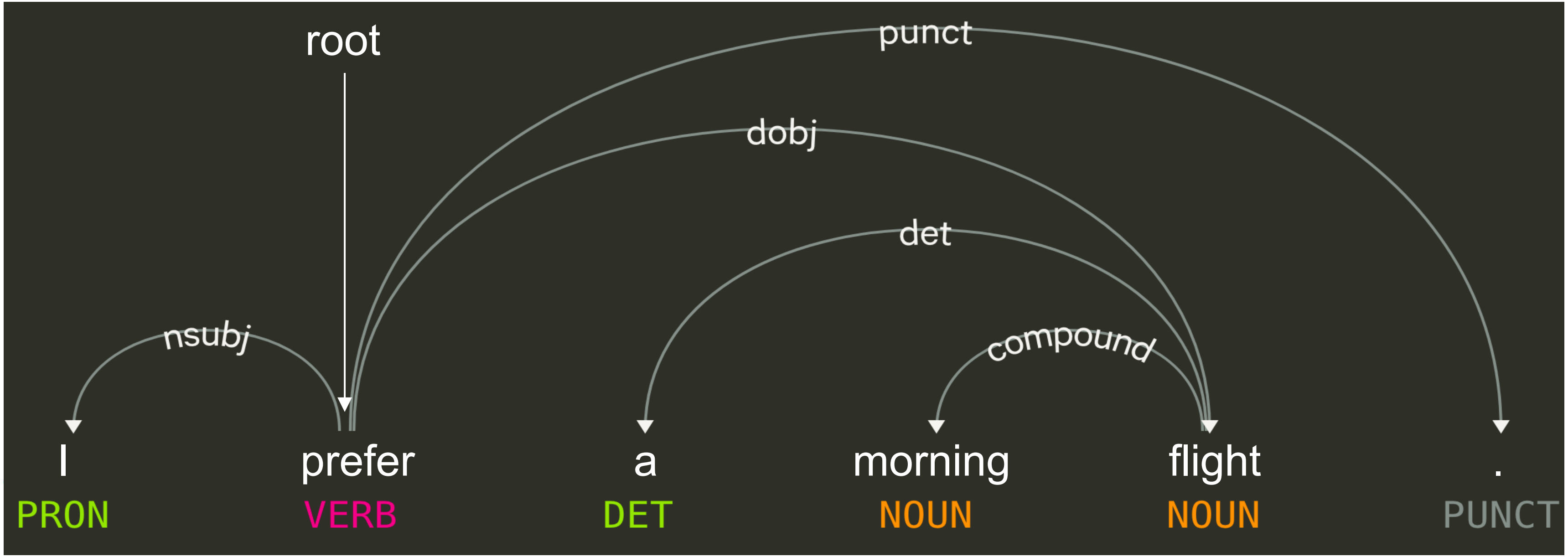
Event extraction

Chunking

- Chunking splits a sentence into phrases, e.g.:
  - [NP *The morning flight*] [PP *from*] [NP *Denver*] [VP *has arrived*.]
  - Chunk tags are useful features for NER
  - https://www.nltk.org/api/nltk.chunk.html
- But chunks don't represent the whole structure of the sentence, so don't give us all we need for tasks like relation extraction.
- Chunking is not suitable for many languages:
  - When phrases can be embedded in one another (e.g., German)
  - If word order is very variable (e.g., Czech, Persian)

Dependency Parsing

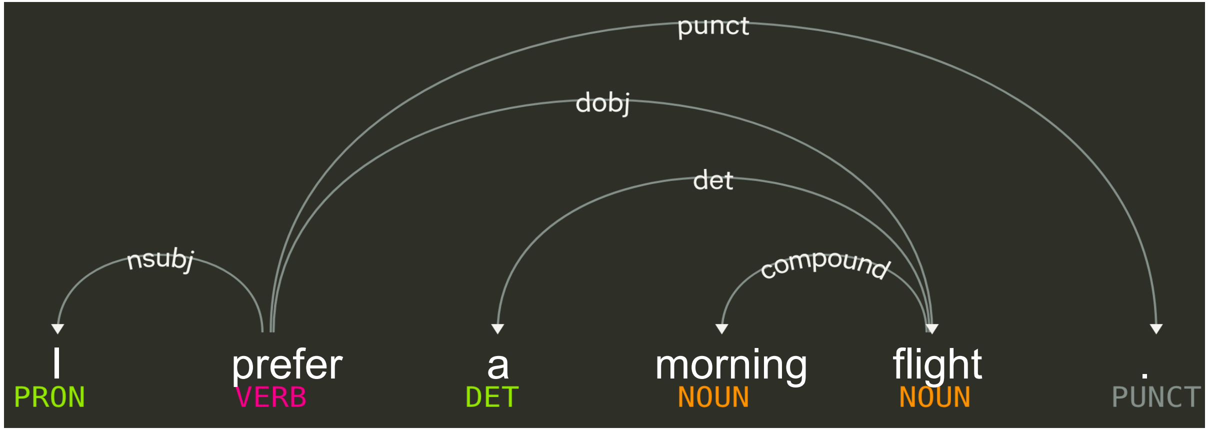
- Extract a graph representing the sentence structure
- https://explosion.ai/demos/displacy



Dependency Parsing

- *prefer* → nsubj → *I*
- *prefer* → dobj → *flight*
- *flight* → det -> *a*
- *flight* → compound → *morning*
- *prefer* → punct → *.*

Each word has one incoming edge and one ancestor, called the *head*



Relations: Verbs

Relation type	Description	Sentence	Relation triple
NSUBJ	Nominal subject	**I** *prefer* a morning flight.	*prefer* → nsubj → *I*
DOBJ	Direct object	I *prefer* a morning **flight**.	*prefer* → dobj → *flight*
IOBJ	Indirect object	We *booked* **her** a flight.	*booked*→ iobj → *her*

Subset of relations from: Nivre, J., et al. (2016). Universal Dependencies v1: A multilingual treebank collection. LREC

Relations: Modifiers

Relation type	Description	Sentence	Relation triple
NMOD	Nominal modifier	I prefer a **morning** *flight.*	*flight* → nmod → *morning*
AMOD	Adjectival modifier	Book the **quickest** *train*.	*train* → amod → *quickest*
NUMMOD	Numeric modifier	They booked **4** *seats*.	*seats* → nummod → *4*

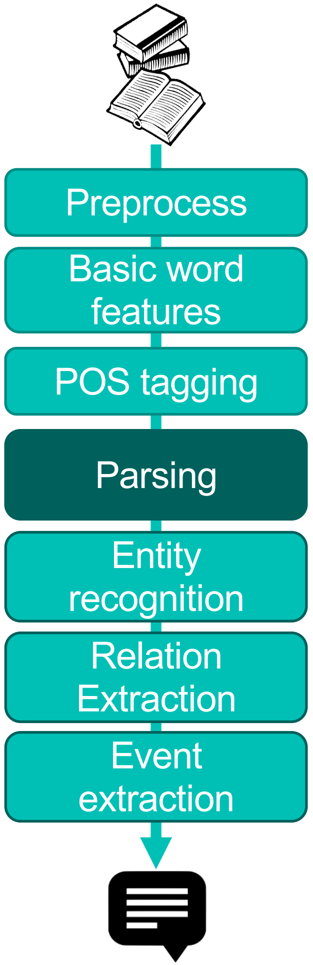
Subset of relations from: Nivre, J., et al. (2016). Universal Dependencies v1: A multilingual treebank collection. LREC

Advantages of Dependency Parsing

- Dependency parsing is effective on languages with variable (free) word order
- Dependencies between verbs and their subjects and objects help identify relations between entities in information extraction
- Modifiers change the meaning of words, e.g., “morning flight” makes the word “flight” more specific
- Dependencies therefore encode vital information for answering questions, such as “which flight does the customer prefer?”

From Parse Trees to Features

- Choice of features is highly task-specific
- Features for a word, *w*:
  - Ancestor word (head)
  - Type of the incoming dependency
- To classify relations between two entities, features may include:
  - Head word of each entity (highest word in the parse tree)
  - Dependency types along the path from one head to another
  - Dependency path length
  - Distance to common ancestor



Challenges of Dependency Parsing

- Goal: Find the dependency relations that make up the sentence
- What are the main challenges of this task?

Ambiguity: multiple possible parses for a sentence.

Fluidity of grammar: sentences that break the rules!

Summary

- Understanding syntax helps us process the meaning of sentences
- Dependency parsing produces a graph of relations between words
- Dependency relations provide features for downstream tasks
- For example, relations between verbs and their subjects and objects help us identify relations between entities.