Syntax

Natural Language Processing and Information Extraction, 2024WS

Lecture 6, 11/29/2024

Gábor Recski

This material can be downloaded from

https://github.com/tuw-nlp-ie/tuw-nlp-ie-2023WS (https://github.com/tuw-nlp-ie/tuw-nlp-ie-2024WS)

Topics and SLP3 chapters

- Parts-of-speech: Chapter 17 (https://web.stanford.edu/~jurafsky/slp3/17.pdf)
- Constituency: Chapter 18 (https://web.stanford.edu/~jurafsky/slp3/18.pdf)
- Dependency: Chapter 19 (https://web.stanford.edu/~jurafsky/slp3/19.pdf)

Dependencies

To run this notebook, you will need to install the **stanza** and **spacy** python packages.

Make sure to restart the kernel afterwards.

Then you can use the cells below to download and initialize the necessary models.

Download models, initialize pipelines

In [1]:
```
import stanza
stanza.download('en')
stanza_nlp = stanza.Pipeline(lang='en', logging_level='WARNING')
```

```
2024-11-28 15:57:14 INFO: Downloading default packages for language: en (Engli
sh) ...


2024-11-28 15:57:15 INFO: File exists: /home/recski/stanza_resources/en/defaul
t.zip
2024-11-28 15:57:20 INFO: Finished downloading models and saved to /home/recsk
i/stanza_resources.
```

In [2]:

```
import spacy
from spacy.cli import download as spacy_download
spacy_download('en_core_web_sm')
spacy_nlp = spacy.load("en_core_web_sm")
```

```
Requirement already satisfied: en-core-web-sm==3.7.1 from https://github.com/e
xplosion/spacy-models/releases/download/en_core_web_sm-3.7.1/en_core_web_sm-3.
7.1-py3-none-any.whl in /home/recski/miniconda3/envs/nlp_course/lib/python3.7/
site-packages (3.7.1)
Requirement already satisfied: spacy<3.8.0,>=3.7.2 in /home/recski/miniconda3/
envs/nlp_course/lib/python3.7/site-packages (from en-core-web-sm==3.7.1) (3.7.
2)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /home/recski/min
iconda3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2-
>en-core-web-sm==3.7.1) (3.0.12)
Requirement already satisfied: jinja2 in /home/recski/miniconda3/envs/nlp_cour
se/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.
1) (3.0.3)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /home/recski/min
iconda3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2-
>en-core-web-sm==3.7.1) (1.0.1)
Requirement already satisfied: typing-extensions<4.5.0,>=3.7.4.1; python_versi
on < "3.8" in /home/recski/miniconda3/envs/nlp_course/lib/python3.7/site-packa
ges (from spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.10.0.2)
Requirement already satisfied: typer<0.10.0,>=0.3.0 in /home/recski/miniconda
3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-co
re-web-sm==3.7.1) (0.4.0)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /home/recski/minicon
da3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-
core-web-sm==3.7.1) (2.0.6)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /home/recski/miniconda3/
envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core
-web-sm==3.7.1) (2.0.3)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /home/recski/minicon
da3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-
core-web-sm==3.7.1) (2.31.0)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /home/recski/minic
onda3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->e
n-core-web-sm==3.7.1) (1.0.2)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /home/recski/miniconda3/
```

```
envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core
-web-sm==3.7.1) (2.4.8)
Requirement already satisfied: langcodes<4.0.0,>=3.2.0 in /home/recski/minicon
da3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-
core-web-sm==3.7.1) (3.3.0)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /home/recski/miniconda
3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-co
re-web-sm==3.7.1) (3.0.2)
Requirement already satisfied: smart-open<7.0.0,>=5.2.1 in /home/recski/minico
nda3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en
-core-web-sm==3.7.1) (5.2.1)
Requirement already satisfied: setuptools in /home/recski/miniconda3/envs/nlp_
course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core-web-sm==
3.7.1) (50.3.0.post20201006)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /home/r
ecski/miniconda3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.
0,>=3.7.2->en-core-web-sm==3.7.1) (1.8.2)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /home/recski/miniconda
3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-co
re-web-sm==3.7.1) (1.1.2)
Requirement already satisfied: weasel<0.4.0,>=0.1.0 in /home/recski/miniconda
3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-co
re-web-sm==3.7.1) (0.3.2)
Requirement already satisfied: numpy>=1.15.0; python_version < "3.9" in /home/
recski/miniconda3/envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.
0,>=3.7.2->en-core-web-sm==3.7.1) (1.21.5)
Requirement already satisfied: packaging>=20.0 in /home/recski/miniconda3/env
s/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core-we
b-sm==3.7.1) (21.3)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /home/recski/miniconda3/
envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core
-web-sm==3.7.1) (4.50.2)
Requirement already satisfied: thinc<8.3.0,>=8.1.8 in /home/recski/miniconda3/
envs/nlp_course/lib/python3.7/site-packages (from spacy<3.8.0,>=3.7.2->en-core
-web-sm==3.7.1) (8.2.1)
Requirement already satisfied: MarkupSafe>=2.0 in /home/recski/miniconda3/env
```

```
s/nlp_course/lib/python3.7/site-packages (from jinja2->spacy<3.8.0,>=3.7.2->en
-core-web-sm==3.7.1) (2.1.1)
Requirement already satisfied: click<9.0.0,>=7.1.1 in /home/recski/miniconda3/
envs/nlp_course/lib/python3.7/site-packages (from typer<0.10.0,>=0.3.0->spacy<
3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (7.1.2)
Requirement already satisfied: zipp>=0.5; python_version < "3.8" in /home/recs
ki/miniconda3/envs/nlp_course/lib/python3.7/site-packages (from catalogue<2.1.
0,>=2.0.6->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.8.0)
Requirement already satisfied: idna<4,>=2.5 in /home/recski/miniconda3/envs/nl
p_course/lib/python3.7/site-packages (from requests<3.0.0,>=2.13.0->spacy<3.8.
0,>=3.7.2->en-core-web-sm==3.7.1) (3.4)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/recski/minico
nda3/envs/nlp_course/lib/python3.7/site-packages (from requests<3.0.0,>=2.13.0
->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.2.0)
Requirement already satisfied: certifi>=2017.4.17 in /home/recski/miniconda3/e
nvs/nlp_course/lib/python3.7/site-packages (from requests<3.0.0,>=2.13.0->spac
y<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2023.7.22)
Requirement already satisfied: urllib3<3,>=1.21.1 in /home/recski/miniconda3/e
nvs/nlp_course/lib/python3.7/site-packages (from requests<3.0.0,>=2.13.0->spac
y<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (2.0.5)
Requirement already satisfied: cloudpathlib<0.16.0,>=0.7.0 in /home/recski/min
iconda3/envs/nlp_course/lib/python3.7/site-packages (from weasel<0.4.0,>=0.1.0
->spacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.15.1)
Requirement already satisfied: confection<0.2.0,>=0.0.4 in /home/recski/minico
nda3/envs/nlp_course/lib/python3.7/site-packages (from weasel<0.4.0,>=0.1.0->s
pacy<3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.1.3)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /home/recski/minico
nda3/envs/nlp_course/lib/python3.7/site-packages (from packaging>=20.0->spacy<
3.8.0,>=3.7.2->en-core-web-sm==3.7.1) (3.0.4)
Requirement already satisfied: blis<0.8.0,>=0.7.8 in /home/recski/miniconda3/e
nvs/nlp_course/lib/python3.7/site-packages (from thinc<8.3.0,>=8.1.8->spacy<3.
8.0,>=3.7.2->en-core-web-sm==3.7.1) (0.7.11)
Requirement already satisfied: importlib_metadata; python_version < "3.8" in /
home/recski/miniconda3/envs/nlp_course/lib/python3.7/site-packages (from cloud
pathlib<0.16.0,>=0.7.0->weasel<0.4.0,>=0.1.0->spacy<3.8.0,>=3.7.2->en-core-web
-sm==3.7.1) (4.11.3)
```

```
✔ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
```

Recap

Tokenization, lemmatization, decompounding

In [3]:
```
doc = stanza_nlp("Did you get me those muffins?")
print("\n".join([f"{word.text:<8}\t{word.lemma}" for word in doc.sentences[0].w
ords]))
```

```
Did             do
you             you
get             get
me              I
those           that
muffins         muffin
?               ?
```

What's next?

```
Twas brillig, and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.
```

(Lewis Carroll: Jabberwocky (https://en.wikipedia.org/wiki/Jabberwocky))

```
Es brillig war. Die schlichten Toven
Wirrten und wimmelten in Waben;
Und aller-mümsige Burggoven
Die mohmen Räth' ausgraben.
```

(Translated by Robert Scott)

They don't make much sense, but how come they make any?

Part-of-speech (POS)

In [4]:
```
print("\n".join([f"{word.text:<8}\t{word.pos}" for word in doc.sentences[0].words]))
```

```
Did             AUX
you             PRON
get             VERB
me              PRON
those           DET
muffins         NOUN
?               PUNCT
```

In [5]:
```
print("\n".join([f"{word.text:<8}\t{word.xpos}" for word in doc.sentences[0].words]))
```

```
Did             VBD
you             PRP
get             VB
me              PRP
those           DT
muffins         NNS
?               .
```

POS-tags are morphosyntactic categories

Word	UPOS (https://universaldependencies.org/u/pos/)		PTB (https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html)
Did	AUX	auxiliary	VBD
you	PRON	pronoun	PRP
get	VERB	verb	VB
me	PRON	pronoun	PRP
those	DET	determiner	DT
muffins	NOUN	noun	NNS
?	PUNCT	punctuation	.

There's always more morphosyntactic features to consider:

In [6]:
```
print("\n".join([f"{word.text:<8}\t{word.pos:<8}\t{word.feats}" for word in doc.sentences[0].words]))
```

```
Did             AUX             Mood=Ind|Number=Sing|Person=2|Tense=Past|VerbF
orm=Fin
you             PRON            Case=Nom|Person=2|PronType=Prs
get             VERB            VerbForm=Inf
me              PRON            Case=Acc|Number=Sing|Person=1|PronType=Prs
those           DET             Number=Plur|PronType=Dem
muffins         NOUN            Number=Plur
?               PUNCT           None
```

Difficulties of POS-tagging

*earnings growth took a **back/JJ** seat*

*a small building in the **back/NN***

*a clear majority of senators **back/VBP** the bill*

*Dave began to **back/VB** toward the door*

*enable the country to buy **back/RP** debt*

*I was twenty-one **back/RB** then*

Chapter 17 (https://web.stanford.edu/~jurafsky/slp3/17.pdf)

Why not implement grammar?

- grammar and vocabulary change too fast
- resolving ambiguities requires probabilistic reasoning

Time	*flies*	*like*	*an*	*arrow*
NOUN	VERB	ADP	DET	NOUN

Time	*flies*	*like*	*an*	*arrow*
VERB	NOUN	ADP	DET	NOUN

Time	*flies*	*like*	*an*	*arrow*
NOUN	NOUN	VERB	DET	NOUN

BTW: the second one can still have three interpretations - can you think of all of them (without googling)?

Questions?

See the supplementary material in `06b_POS_tagging_HMMs.ipynb` on POS-tagging with Hidden Markov Models

Syntactic structure

Two perspectives

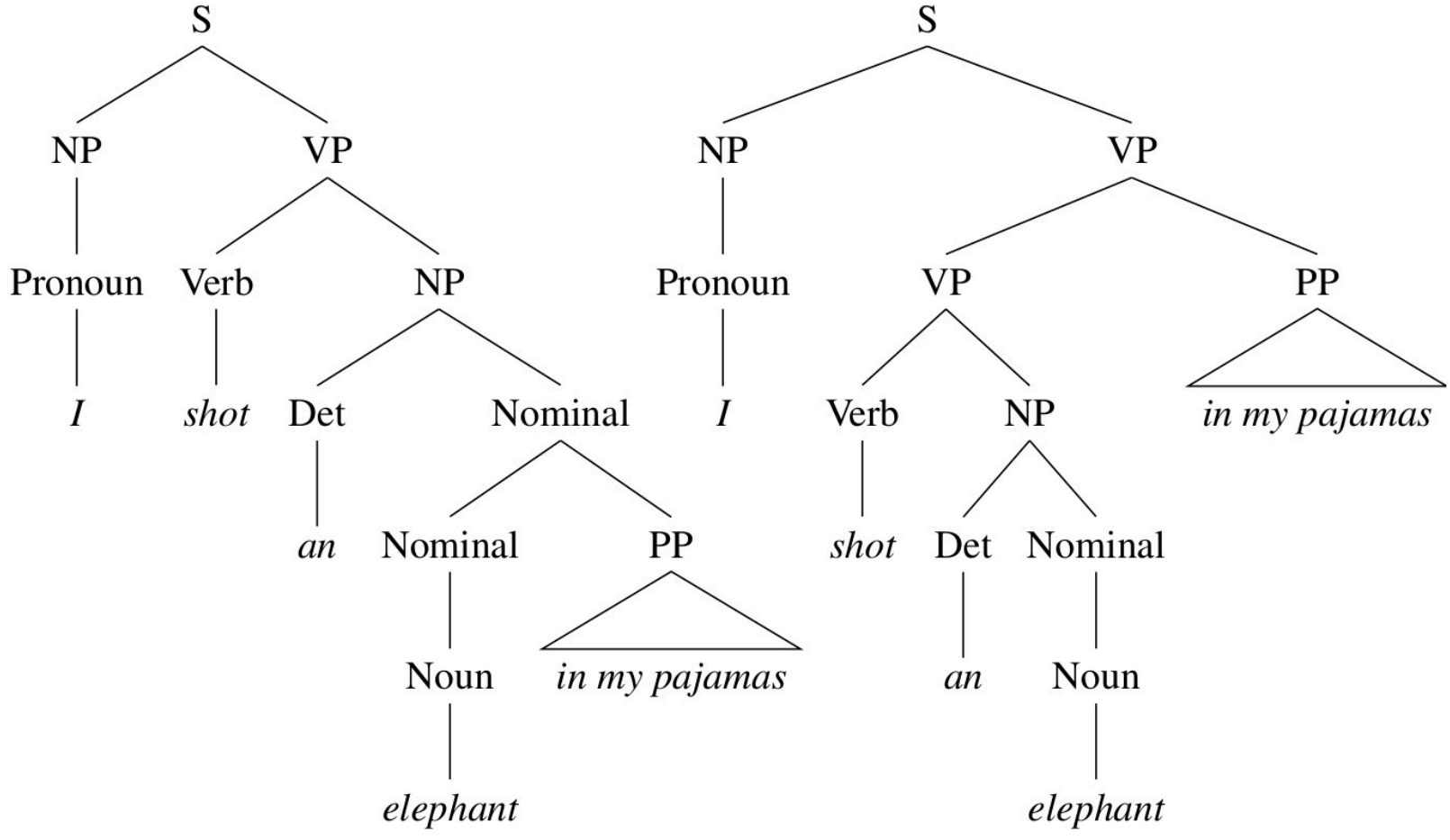
- Constituency structure
- Dependency structure

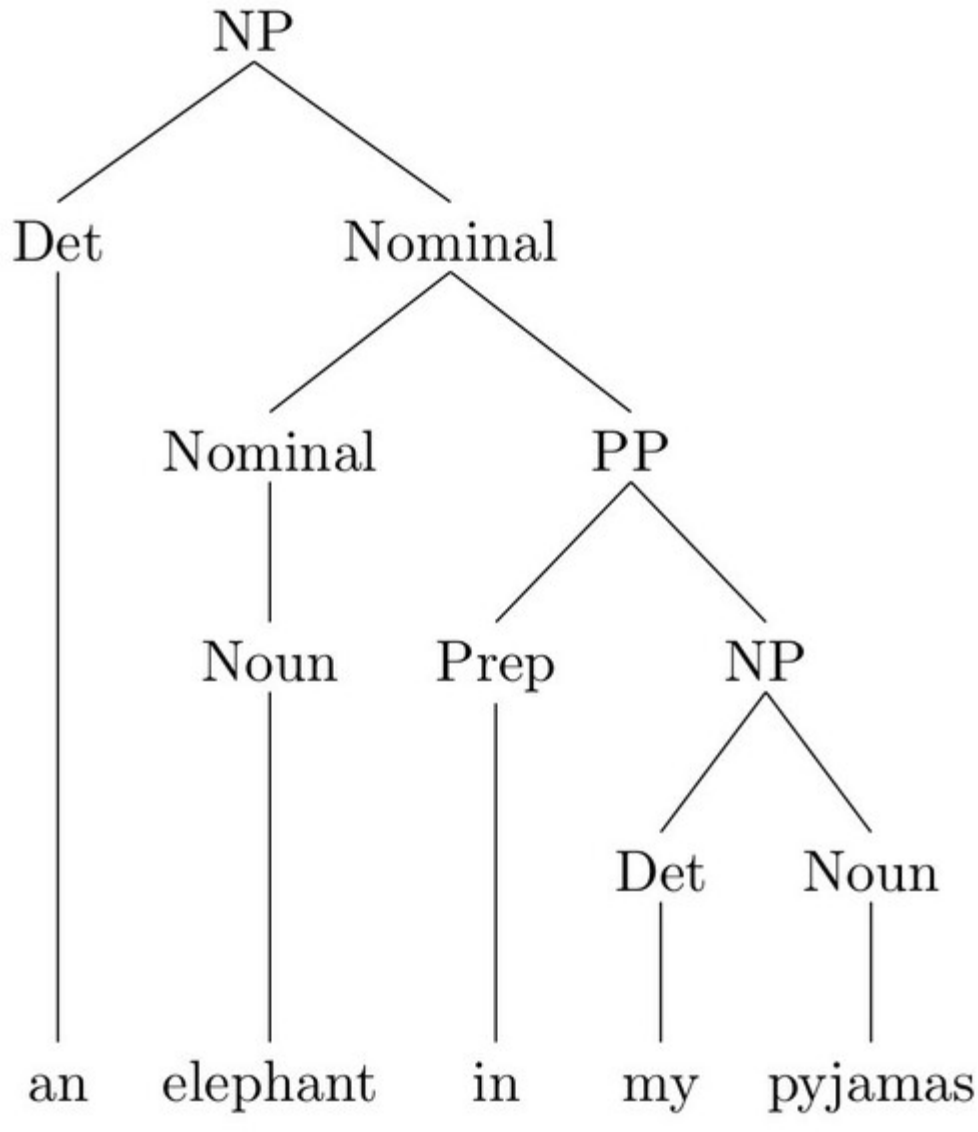
Constituency

I shot an elephant in my pyjamas

In [7]:
```
doc = stanza_nlp("I shot an elephant in my pyjamas")
print("\n".join([f"{word.text:<12}{word.pos}" for word in doc.sentences[0].word
s]))
```

```
I           PRON
shot        VERB
an          DET
elephant    NOUN
in          ADP
my          PRON
pyjamas     NOUN
```


S
NP
VP
Pronoun
Verb
NP
I
shot
Det
Nominal
an
Nominal
PP
Noun
in my pajamas
elephant
S
NP
VP
Pronoun
VP
PP
I
Verb
NP
in my pajamas
shot
Det
Nominal
an
Noun
elephant


NP
Det
Nominal
Nominal
PP
Noun
Prep
NP
Det
Noun
an
elephant
in
my
pyjamas

```
(NP
    (DET an)
    (Nominal
        (Nominal
            (NOUN elephant)
        )
    (PP
        (PREP in)
        (NP
            (DET my)
            (NOUN pyjamas)
        )
    )
)
```

NP, PP, etc. are distributional categories. Just like POS-tags!

(DET an) (NOUN elephant) (PREP in) (DET my) (NOUN pyjamas)

(DET two) (NOUN pandas) (PREP behind) (DET his) (NOUN tent)

(NP I) (VERB shot) (NP an elephant) (PP in my pyjamas)

(NP My best friend) (VERB met) (NP two pandas) (PP behind his tent)

(NP I) (VP shot an elephant in my pyjamas)

(NP The guy driving the jeep) (VP fainted)

Phrase structure grammars

```
S -> NP VP
VP -> VERB (NP)
NP -> (DET) NOUN (PP)
PP -> PREP NP
(...)
DET -> (an|the|my|his|...)
VERB -> (shot|met|fainted...)
PREP -> (in|behind|...)
NOUN -> (I|elephant|pyjamas|panda|tent|jeep|guy|...)
```

Probabilistic grammars

```
NOUN -> I (0.8)
NOUN -> elephant (0.1)
(...)
VP -> VERB (0.2)
VP -> VERB NP (0.8)
```

Constituency parsing

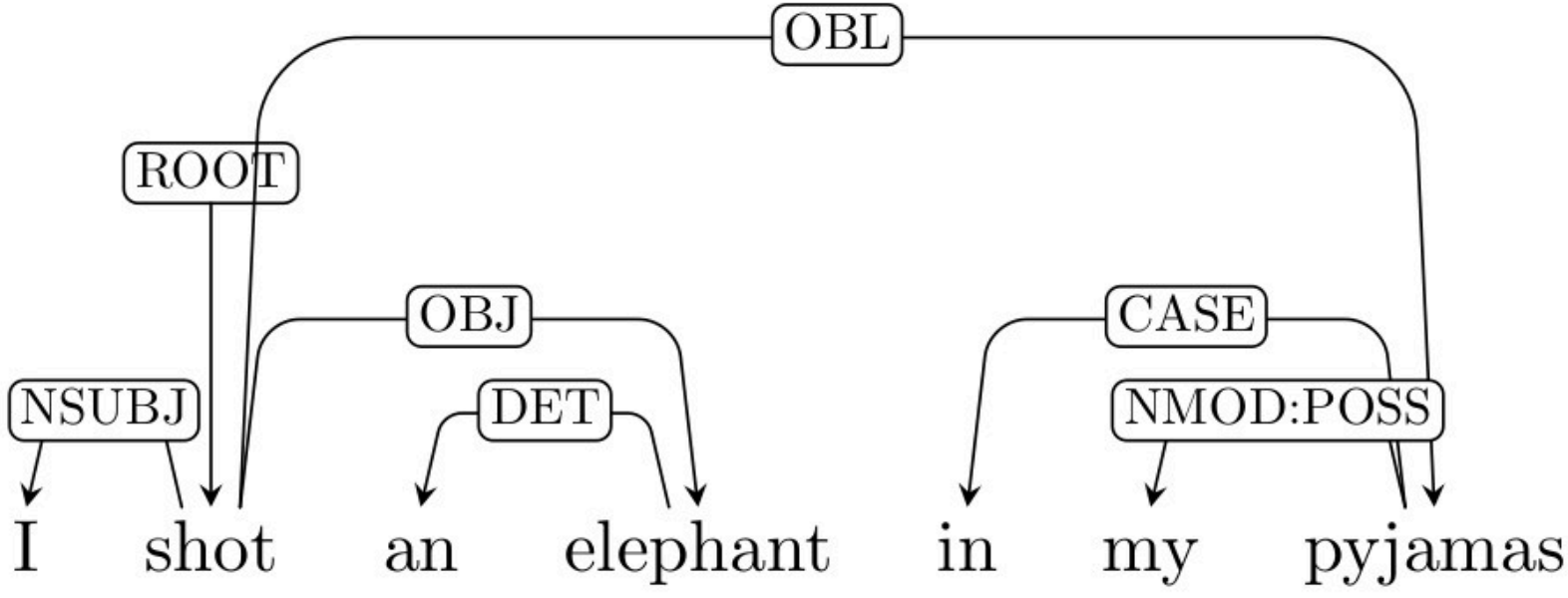
Parsing is the task of determining the (most likely) possible derivations of a sentence, given a (probabilistic) grammar

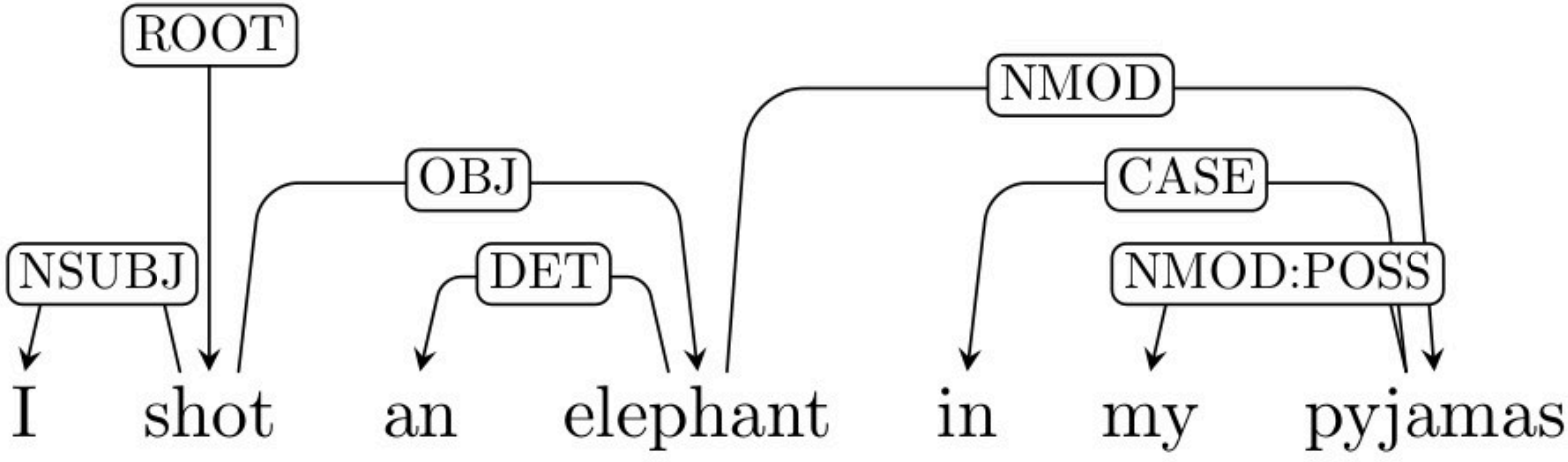
The CKY algorithm

See example in cky.pdf (cky.pdf)

Questions?

Dependency structure


OBL
ROOT
OBJ
CASE
NSUBJ
DET
NMOD:POSS
I
shot
an
elephant
in
my
pyjamas


ROOT
NMOD
OBJ
CASE
NSUBJ
DET
NMOD:POSS
I
shot
an
elephant
in
my
pyjamas

- **NSUBJ**: nominal subject
- **OBJ**: object
- **DET**: determiner
- **OBL**: oblique nominal
- **NMOD**: nominal modifier
- **POSS**: possessive

In [8]:

```
doc = stanza_nlp("I shot an elephant in my pyjamas")
print("\n".join([f"{word.id:<4}{word.text:<12}{word.deprel:<12}{word.head:<8}"
for word in doc.sentences[0].words]))
```

```
1   I           nsubj       2
2   shot        root        0
3   an          det         4
4   elephant    obj         2
5   in          case        7
6   my          nmod:poss   7
7   pyjamas     obl         2
```

Dependency parsing - approaches

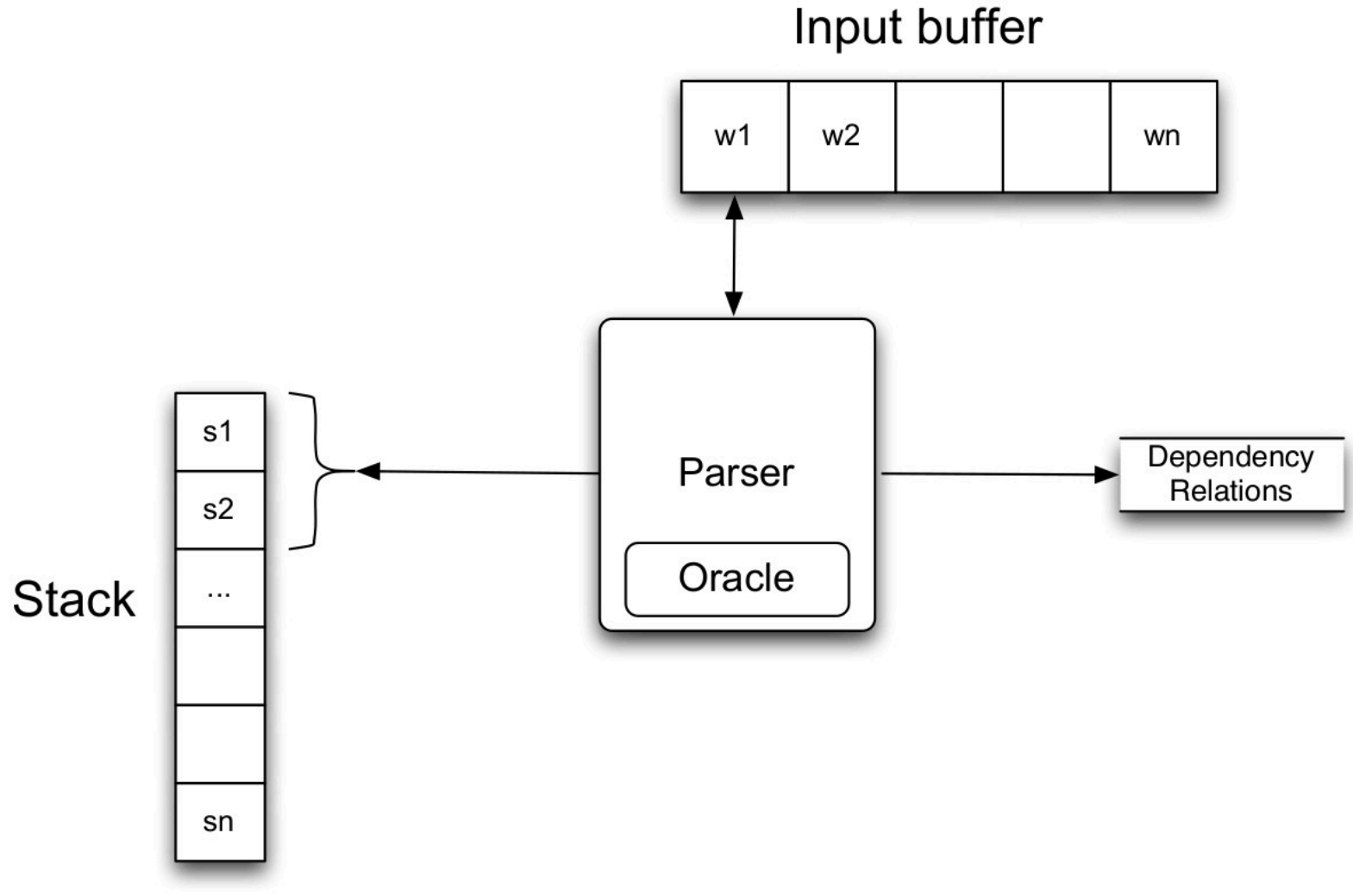
Arc-factored parsing

- model the likelihood of edges
- e.g. how likely is *nmod(elephant, pyjamas)*?
- find the dependency graph with the most likely edges

Transition-based parsing

- build dependency graphs by adding one word at a time
- model the likelihood of possible next steps
- e.g. should I attach *pyjamas* to *elephant* or *shot*?

Shift-reduce parsing


Input buffer
w1
w2
wn
s1
s2
...
sn
Stack
Parser
Oracle
Dependency Relations

Shift-reduce parsing

- transition-based approach
- processes words one-by-one, in linear order, no backtracking
- for each word, choose between:
    - **shift**: push the next word on the **stack**
    - **reduce**: add a dependency edge between the top two words on the stack, and remove the dependent.

Shift-reduce example

See shiftreduce.pdf (shiftreduce.pdf)

A historical note on the two perspectives

Constituency structure

- Origins in **structural linguistics** (F. de Saussure, 1900s and later L. Bloomfield, 1930s)
- (The basic ideas actually date back to **Pāṇini** (~500 BCE))
- Application of **formal language theory** (e.g. PS grammars) in 1950s (N. Chomsky)
- Remains the mainstream perspective in theoretical linguistics (known as **generative grammar**)

Dependency structure

- Origins in **Dependency grammar** (Tesnière, 1950s)
- (The basic ideas actually date back to **Pāṇini** (~500 BCE))
- Widespread use in NLP

Questions?