Lecture 18: FST, Part-of-Speech Tagging

Ling 1330/2330 Intro to Computational Linguistics
Na-Rae Han, 10/31/2023

Outline

- HW6 review, FST wrap-up

- Part-of-speech tagging
  - *Language and Computers*, Ch. 3.4 Tokenization, POS tagging
  - NLTK Book Ch.5 Categorizing and tagging words
- Parts of speech
- POS ambiguity
- POS-tagged corpora
- POS tagging practice

Homework 6: LEXC script

english.lexc

```
Multichar_Symbols +N +V +PastPart +Past +PresPart +3P +Sg +Pl
+ADJ +Comp +Supl +ADV       ! new multi-char symbols for PART 2

LEXICON Root
Noun ;
Verb ;
Adjective ;

LEXICON Noun
cat   Ninf;
city  Ninf;
mouse   Ninf;       ! 'mouses' will be overridden in .foma
child   Ninf;       ! 'childs' will be overridden in .foma
cactus  Ninf;       ! parallel form 'cacti' will be added in .foma
octopus Ninf;       ! parallel form 'octopi' will be added in .foma

LEXICON Verb
beg   Vinf;
fox   Vinf;
make  Vinf;
…
```

Homework 6: LEXC script

english.lexc

```
study   Vinf;
stop    Vinf;
nod     Vinf;
fold    Vinf;
write   Vinf;
drink   Vinf;
sneak   Vinf;

LEXICON Ninf
+N+Sg:0   #;
+N+Pl:^s  #;

LEXICON Vinf
+V:0               #;
+V+3P+Sg:^s        #;
+V+Past:^ed        #;
+V+PastPart:^ed    #;
+V+PresPart:^ing   #;
^able:^able        ADJinf2;
```

```
LEXICON Adjective
slow    ADJinf;
cool    ADJinf;
fine    ADJinf;
beautiful   ADJinf2;
excellent   ADJinf2;

LEXICON ADJinf
ADJinf2;
+ADJ+Comp:^er      #;
+ADJ+Supl:^est     #;

LEXICON ADJinf2
+ADJ:0      #;
^ly:^ly     ADVinf;

LEXICON ADVinf
+ADV:0      #;
```

Upper level = Lower level, could have just used ^ly

Homework 6: FOMA script

english.foma

```
### english.foma ###

# Vowels
define V [ a | e | i | o | u ];

# Consonant doubling: 1-letter consonant doubled before -ing/-ed (beg/begging)
# Add p, d, and also constrain context with V (needed for *foldded)
define ConsonantDoubling g -> g g, p -> p p, d -> d d || V _ "^" [i n g|e d|a b l e];

# E deletion: silent e dropped before -ing and -ed (make/making)
define EDeletion e -> 0 || _ "^" [i n g | e d | e r | e s t | a b l e] ;

# E insertion e added after -s, -z, -x, -ch, -sh before s (watch/watches)
define EInsertion [..] -> e || s | z | x | c h | s h _ "^" s ;


…
```

English morphology as FST

- "english.lexc"
  - contains noun, verb and adjective stems.
  - implements **morphotactic grammar** as a network of continuation classes.
  - implements morpheme tags (+N, +Past, etc.) and their suffix representation.
- "english.foma"
  - implements **alternation rules** as individual FSTs.
  - It also implements irregular forms and parallel forms as individual FSTs.
  - It then composes the entire FST out of the initial lexicon and the rule FSTs.

Your morphotactic grammar (lexc) and alternation rules (foma) should be set up as a natural reflection of English morphology!

→ Expansion and upkeep become more intuitive.

FST morphology development, IRL

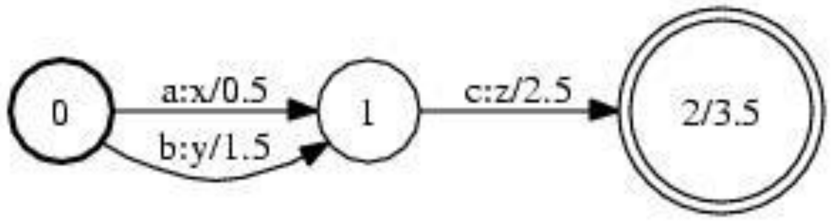
- A Finnish FST example
- A Korean FST… by a certain individual

FSTs: advantages

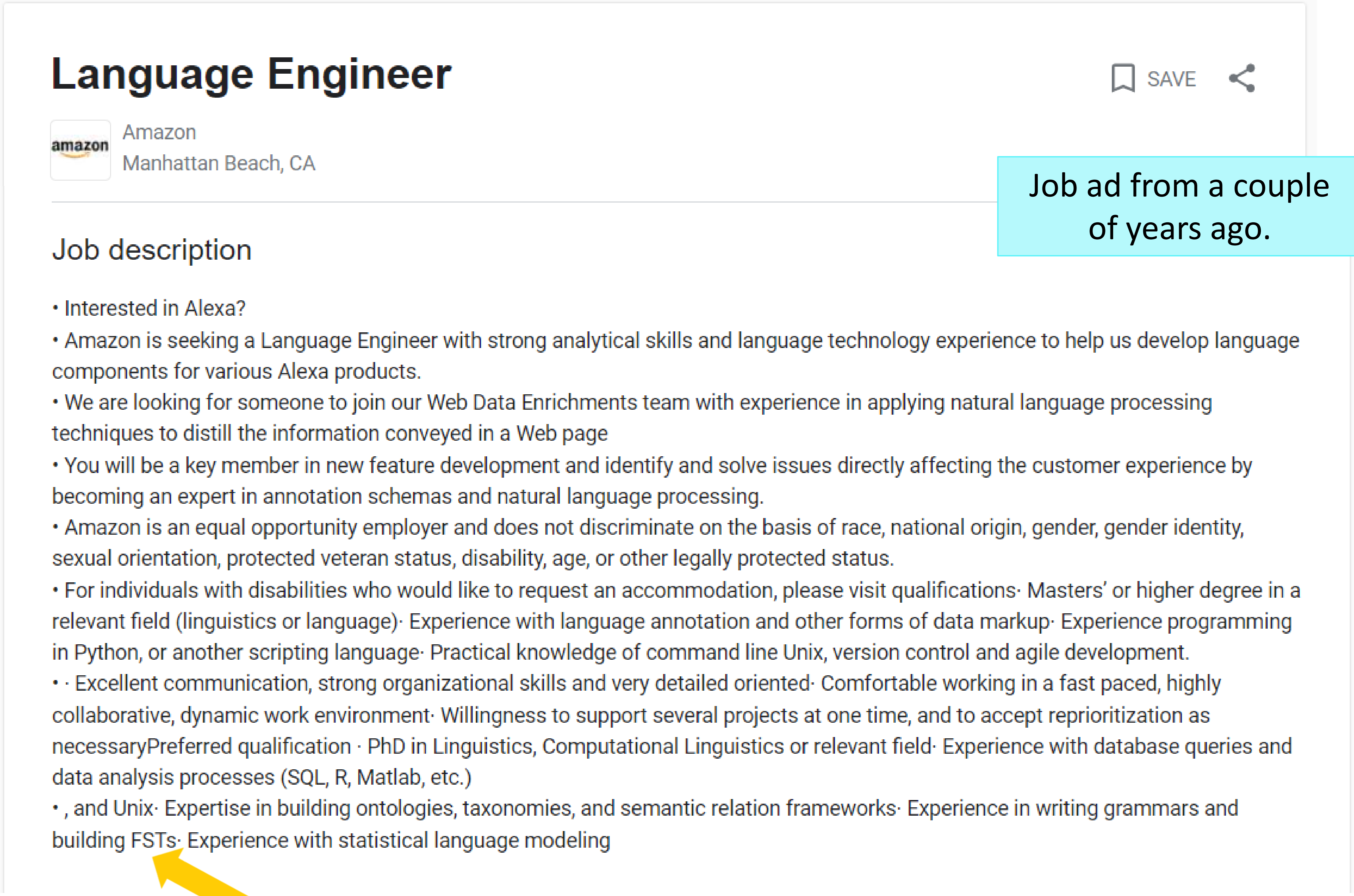
- A single network can be used as a morphological analyzer AND a generator.
- FST operations are fast, efficient, and computationally elegant.
- FST operations (concatenation, composition, union, subtraction, …) are algorithmically rigorous with well-understood mathematical properties.
- Developing a FST-based lexicon is intuitive:
  - Separate FST networks can be developed in isolation that target a specific linguistic phenomenon ("e deletion", "consonant doubling", etc.)
  - A collection of which can then be composed into a single FST network
  - Linguists are well-positioned to develop such systems using their knowledge of morpho-syntax and morpho-phonemics.

- Are they a stochastic/statistical/probabilistic NLP system?
  - NO. FST morphology is entirely rule-based.
  - FST morphology is a classic example of **symbolic computational linguistics**. Language is modelled as a formal system which is tightly coupled with computational machinery.

- Marrying FST and probability?
  - Probabilistic FSTs do exist: Open FST (originally developed by AT&T) implements **weighted** FST.



- Are FSTs good for morphology only?
  - NO. FST can be used to implement all aspects of grammar.



Language Engineer

SAVE

Amazon
Manhattan Beach, CA

Job ad from a couple of years ago.

Job description

• Interested in Alexa?
• Amazon is seeking a Language Engineer with strong analytical skills and language technology experience to help us develop language components for various Alexa products.
• We are looking for someone to join our Web Data Enrichments team with experience in applying natural language processing techniques to distill the information conveyed in a Web page
• You will be a key member in new feature development and identify and solve issues directly affecting the customer experience by becoming an expert in annotation schemas and natural language processing.
• Amazon is an equal opportunity employer and does not discriminate on the basis of race, national origin, gender, gender identity, sexual orientation, protected veteran status, disability, age, or other legally protected status.
• For individuals with disabilities who would like to request an accommodation, please visit qualifications· Masters' or higher degree in a relevant field (linguistics or language)· Experience with language annotation and other forms of data markup· Experience programming in Python, or another scripting language· Practical knowledge of command line Unix, version control and agile development.
• · Excellent communication, strong organizational skills and very detailed oriented· Comfortable working in a fast paced, highly collaborative, dynamic work environment· Willingness to support several projects at one time, and to accept reprioritization as necessaryPreferred qualification · PhD in Linguistics, Computational Linguistics or relevant field· Experience with database queries and data analysis processes (SQL, R, Matlab, etc.)
• , and Unix· Expertise in building ontologies, taxonomies, and semantic relation frameworks· Experience in writing grammars and building FSTs· Experience with statistical language modeling

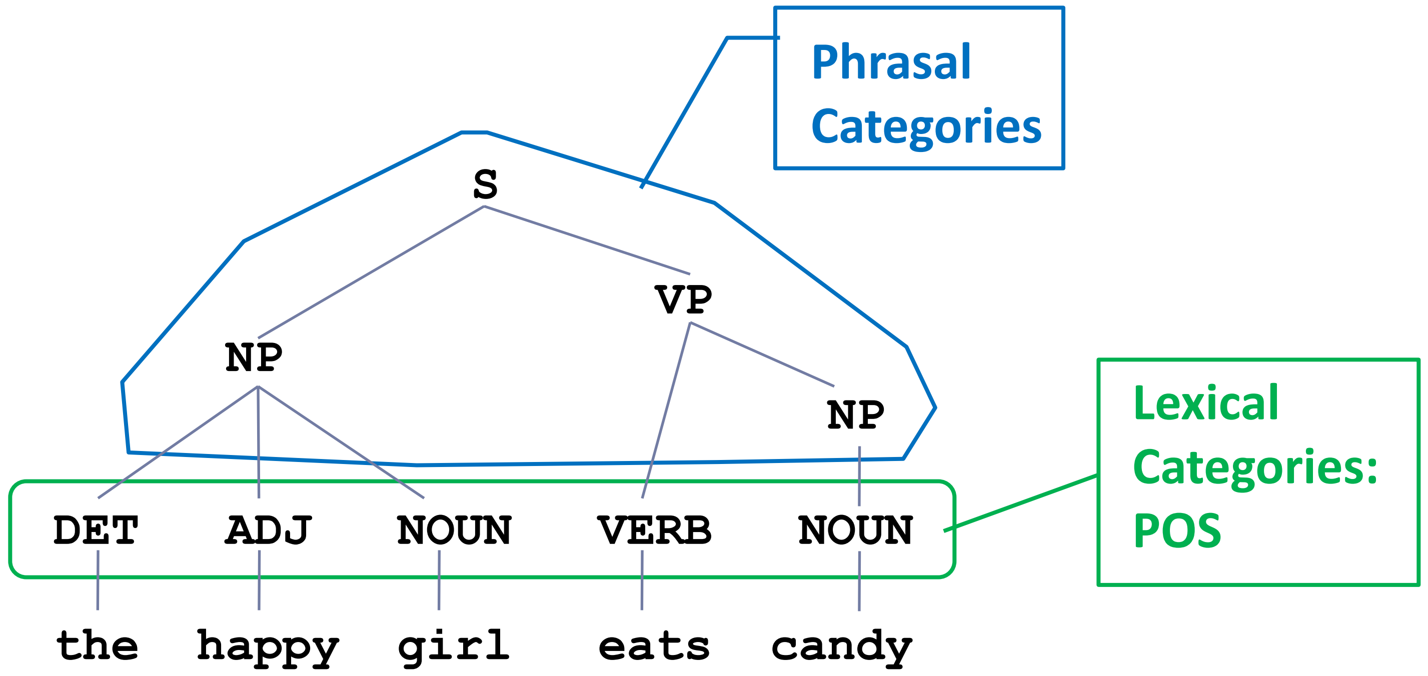
FSA/FST and formal language theory

- FSA/FST can be thought of as a *type* of grammar.
- Question: is FSA/FST adequate as a grammatical framework for English? How about other languages?

🡐 We will return to this question later, when we focus on **formal language theory**.

Phrasal vs. Lexical Categories

- A tree structure for *The happy girl eats candy*:



Part-of-speech

- Also known as: **lexical categories**
- Assigned to <u>individual words</u>

- **NOUN**, **VERB**, **ADJ** (Adjective), **ADV** (Adverb)

  ← Open lexical categories; words have semantic content

- **DET** (Determiner), **ADP** (Adposition/Preposition), **PRON** (Pronoun), **PRT** (Particle), **CONJ** (Conjunction)

  ← Closed lexical categories; grammatical/functional words

* Universal POS tagset.
Also: NUM, ., X

Is That All?

- Interjections
  - *Oh no, that's too bad*
  - *Yes, I'd like that.*
  - *That's nice, eh?*
  - *Hooray, she won gold*

- There are a few odd ones that are hard to classify:
  - *to* in infinitives
    - *I tried to finish.* (cf. *I went to school*)
  - negative particle *not*
    - *She did not eat. She is not happy.*

- And many more.
- Tagsets used in NLP are typically larger and finer-grained.

POS ambiguity

- Some words can take on **multiple parts-of-speech**
  - *Tim likes to go for walks. / Joe walks to school every day.*
  - *January is a cold month. / I have a very bad cold.*
  - *We need more books. / We need a more interesting book.*
  - *I should do my homework. / I did finish my homework.*
  - *I like that pie. / I like that. / I told you that he's lying.*

← Ultimately, their position within a larger phrase (or sentence) must be considered in order to determine their POS

← Part-of-speech tagging must resolve **ambiguity**

← In Brown Corpus, 11.5% of all word types and 40% of word tokens are ambiguous!

Part-of-speech tagging

- POS tagging is a process by which **a single POS tag** is assigned to **each word** (and symbols/punctuations) in a text.

```
The  happy  girl  eats  candy  .
DET   ADJ   NOUN  VERB   NOUN  .
```

- This is one of the earlier steps in the NLP pipeline, following tokenization.

POS-tagged corpora

- Manually POS-tagged large-scale corpora have been instrumental in advancing statistical NLP technologies.

- The Brown Corpus
  - 1 million words (1.16 mil after tokenization)

- The Penn Treebank Corpus
  - 1 million words (1st volume)
  - ←A small portion (100K words) included in NLTK data
  - ←Also has syntactic annotation
  - ←MANY subsequent volumes, in many different languages.

The Brown Corpus

```
The/at Fulton/np-tl County/nn-tl Grand/jj-tl Jury/nn-tl said/vbd
Friday/nr an/at investigation/nn of/in Atlanta's/np$ recent/jj
primary/nn election/nn produced/vbd ``/`` no/at evidence/nn ''/''
that/cs any/dti irregularities/nns took/vbd place/nn ./.

The/at jury/nn further/rbr said/vbd in/in term-end/nn
presentments/nns that/cs the/at City/nn-tl Executive/jj-tl
Committee/nn-tl ,/, which/wdt had/hvd over-all/jj charge/nn of/in
the/at election/nn ,/, ``/`` deserves/vbz the/at praise/nn and/cc
thanks/nns of/in the/at City/nn-tl of/in-tl Atlanta/np-tl ''/''
for/in the/at manner/nn in/in which/wdt the/at election/nn was/bedz
conducted/vbn ./.

The/at September-October/np term/nn jury/nn had/hvd been/ben
charged/vbn by/in Fulton/np-tl Superior/jj-tl Court/nn-tl Judge/nn-tl
Durwood/np Pye/np to/to investigate/vb reports/nns of/in possible/jj
``/`` irregularities/nns ''/'' in/in the/at hard-fought/jj primary/nn
which/wdt was/bedz won/vbn by/in Mayor-nominate/nn-tl Ivan/np
Allen/np Jr./np ./.
```

The Penn Treebank

http://languagelog.ldc.upenn.edu/nll/?p=3594

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
( (S
    (NP-SBJ (NNP Mr.) (NNP Vinken) )
    (VP (VBZ is)
      (NP-PRD
        (NP (NN chairman) )
        (PP (IN of)
          (NP
            (NP (NNP Elsevier) (NNP N.V.) )
            (, ,)
            (NP (DT the) (NNP Dutch) (VBG publishing) (NN group) )))))
    (. .) ))
```

POS tagsets

- There are multiple POS tagsets for English in use.
  - Some are larger, some are smaller.
- **The Brown Corpus tagset** (87 tags)
  - http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM
- In NLP, **the Penn Treebank tagset** (45 tags) has become de facto standard.
  - http://www.surdeanu.info/mihai/teaching/ista555-fall13/readings/PennTreebankTagset.html
  - This is the default tagset for `nltk.pos_tag()`.
- NLTK lets you load a POS-tagged corpus using "**Universal**" **POS tagset** (only 12 tags).
  - https://www.nltk.org/book/ch05.html#a-universal-part-of-speech-tagset

POS-tagged corpora in NLTK

- nltk_data include many corpus resources with POS tags.
  - The Brown Corpus, The Penn Treebank Corpus, NPS Chat Corpus, Chinese, Hindi, Spanish, Portuguese…
  - You can load them using the default tagset (Penn Treebank or Brown) or the Universal POS tagset.

```
>>> from nltk.corpus import brown
>>> brown.tagged_sents()[0]
[('The', 'AT'), ('Fulton', 'NP-TL'), ('County', 'NN-TL'), ('Grand', 'JJ-TL'),
('Jury', 'NN-TL'), ('said', 'VBD'), ('Friday', 'NR'), ('an', 'AT'),
('investigation', 'NN'), ('of', 'IN') …
>>> brown.tagged_sents()[31]
[('His', 'PP$'), ('petition', 'NN'), ('charged', 'VBD'), ('mental', 'JJ'),
('cruelty', 'NN'), ('.', '.')]
>>> brown.tagged_sents(tagset='universal')[31]
[('His', 'DET'), ('petition', 'NOUN'), ('charged', 'VERB'), ('mental', 'ADJ'),
('cruelty', 'NOUN'), ('.', '.')]
```

POS-tagged corpora in NLTK

- You can also load them as POS-tagged *words* or POS-tagged *sentences*.

```
>>> from nltk.corpus import treebank
>>> treebank.tagged_words()[:10]
[('Pierre', 'NNP'), ('Vinken', 'NNP'), (',', ','), ('61', 'CD'), ('years', 'NNS'),
('old', 'JJ'), (',', ','), ('will', 'MD'), ('join', 'VB'), ('the', 'DT')]
>>> treebank.tagged_sents()[9]
[('There', 'EX'), ('is', 'VBZ'), ('no', 'DT'), ('asbestos', 'NN'), ('in', 'IN'),
('our', 'PRP$'), ('products', 'NNS'), ('now', 'RB'), ('.', '.'), ("''", "''")]
>>> treebank.tagged_sents(tagset='universal')[9]
[('There', 'DET'), ('is', 'VERB'), ('no', 'DET'), ('asbestos', 'NOUN'), ('in',
'ADP'), ('our', 'PRON'), ('products', 'NOUN'), ('now', 'ADV'), ('.', '.'), ("''",
'.')]
```

POS annotation practice

10 minutes

- Manually annotate this sentence with POS tags!
  - *Mary had a little lamb, his fleece was white as snow, and everywhere that Mary went, the lamb was sure to go.*

1. Universal POS tagset
2. Penn Treebank POS tagset
3. Brown POS tagset

🡐 Will paste links to 3 tagsets in MS Teams

🡐 Start with Universal, then Brown, (then Penn Treebank if time)

	Universal POS tagset	**Penn Treebank tagset**	**Brown POS tagset**
Mary	NOUN	NNP	NP
had	VERB	VBD	HVD
a	DET	DT	AT
little	ADJ	JJ	JJ
lamb	NOUN	NN	NN
,	.	,	,
his	PRON (or, DET)	PRP$	PP$
fleece	NOUN	NN	NN
was	VERB	VBD	BEDZ
white	ADJ	JJ	JJ
as	ADP	IN	IN
snow	NOUN	NN	NN
,	.	,	,

	Universal POS tagset	**Penn Treebank tagset**	**Brown POS tagset**
and	CONJ	CC	CC
everywhere	ADV (or PRON)	RB	RB
that	ADV or ADP	WDT (relative clause; IN is for non-relative clauses)	WPS (relative clause?; CS is for non-relative clauses)
Mary	NOUN	NNP	NP
went	VERB	VBD	VBD
,	.	,	,
the	DET	DT	AT
lamb	NOUN	NN	NN
was	VERB	VBD	BEDZ
sure	ADJ	JJ	JJ
to	PRT	TO	TO
go	VERB	VB	VB
.	.	.	.

Wrapping up

- Exercise 9 out
  - Explore POS in the Brown corpus
- Thursday
  - More POS
- Nov 16 (Thu) class will be remote, over Zoom.
- Final exam schedule announced!
  - 12/13 (Wed) 4-5:50pm
  - At LMC's PC lab (G17 CL)