LING/C SC 581:

Advanced Computational Linguistics

Lecture 10

Administriva

- Lectures 6-9 are canceled. Sorry.

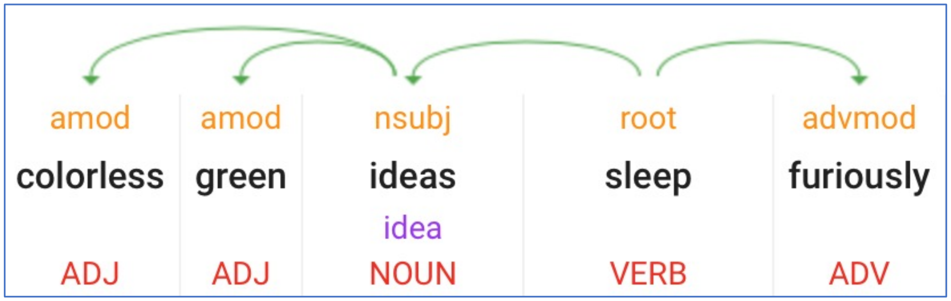
- Let's reboot the course with some syntactic analysis …
- Quick Homework 4 (*do it before next time*)

Universal Dependencies

- We have seen some examples from 538 last semester (*see also next slide*).
- Stanford Dependencies (SD) (de Marneffe and Manning, 2008, rev. 2016)
    - https://downloads.cs.stanford.edu/nlp/software/dependencies_manual.pdf
    - still used by Google Natural Language
- Universal Stanford Dependencies (de Marneffe *et al.*, 2014)
    - https://nlp.stanford.edu/pubs/USD_LREC14_paper_camera_ready.pdf
- Universal Dependencies (UD) v2
    - https://universaldependencies.org/u/dep/index.html
    - "Universal Dependencies (UD) is a project that is developing cross-linguistically consistent treebank annotation for many languages, with the goal of facilitating **multilingual parser development**, cross-lingual learning, and parsing research from a language typology perspective."

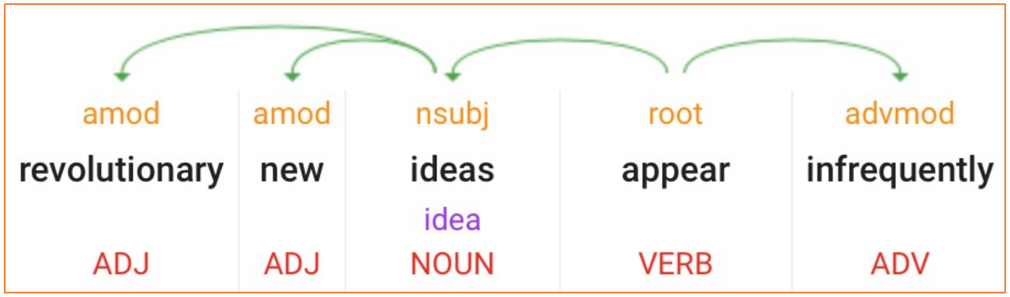
Example

- pg145-146 in *LSLT* (Chomsky 1955/1975)
- https://cloud.google.com/natural-language/



amod	amod	nsubj	root	advmod
colorless	green	ideas	sleep	furiously
		idea		
ADJ	ADJ	NOUN	VERB	ADV

- sequence ADJ ADJ NOUN VERB ADV is grammatical



amod	amod	nsubj	root	advmod
revolutionary	new	ideas	appear	infrequently
		idea		
ADJ	ADJ	NOUN	VERB	ADV

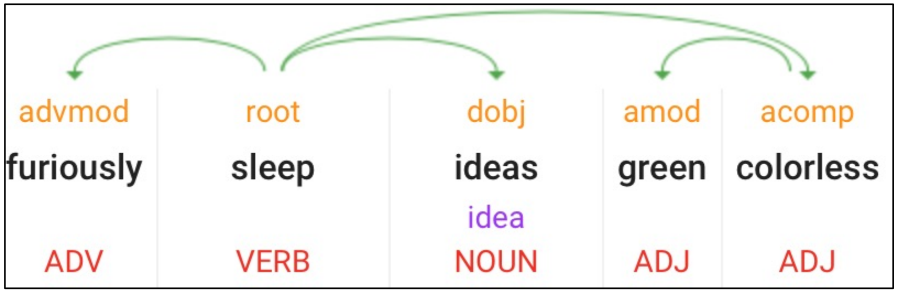
Example

- There's no notion of **grammaticalness**, *Google Natural Language* doesn't rate its inputs (*produces top parse only*)

11 *colorless green ideas sleep furiously*

12 *furiously sleep ideas green colorless*

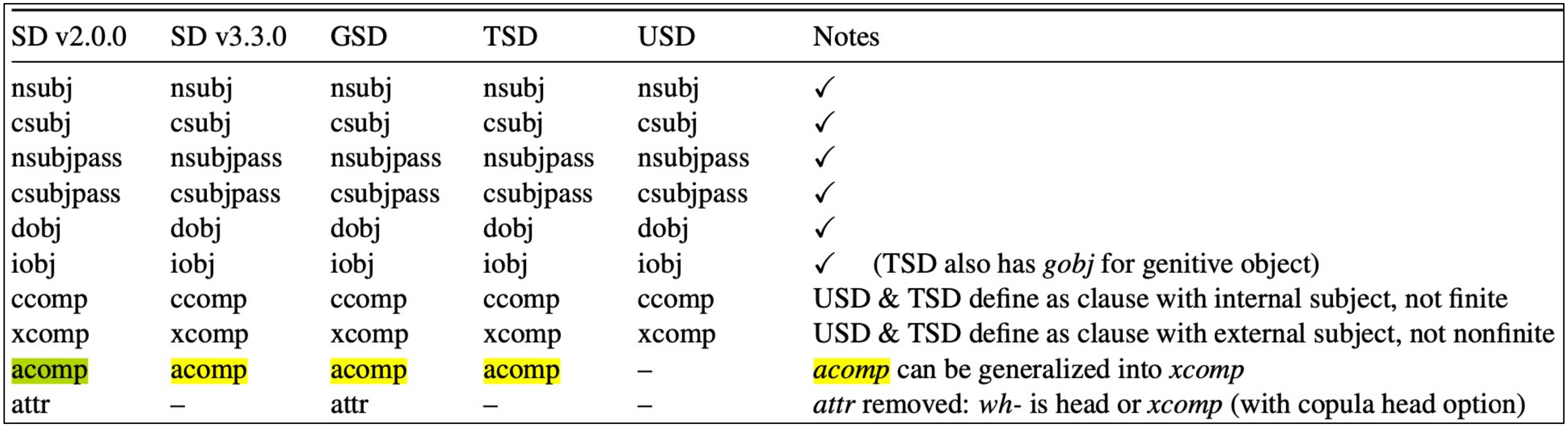
can you parse this?



BTW:
`acomp` not in UD

Example

(de Marneffe *et al.*, 2014)

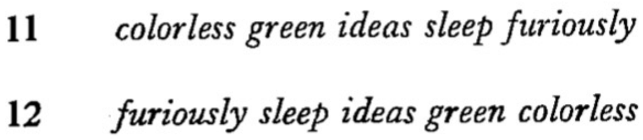


SD v2.0.0	SD v3.3.0	GSD	TSD	USD	Notes
nsubj	nsubj	nsubj	nsubj	nsubj	✓
csubj	csubj	csubj	csubj	csubj	✓
nsubjpass	nsubjpass	nsubjpass	nsubjpass	nsubjpass	✓
csubjpass	csubjpass	csubjpass	csubjpass	csubjpass	✓
dobj	dobj	dobj	dobj	dobj	✓
iobj	iobj	iobj	iobj	iobj	✓ (TSD also has *gobj* for genitive object)
ccomp	ccomp	ccomp	ccomp	ccomp	USD & TSD define as clause with internal subject, not finite
xcomp	xcomp	xcomp	xcomp	xcomp	USD & TSD define as clause with external subject, not nonfinite
acomp	acomp	acomp	acomp	–	*acomp* can be generalized into *xcomp*
attr	–	attr	–	–	*attr* removed: *wh-* is head or *xcomp* (with copula head option)

xcomp: open clausal complement

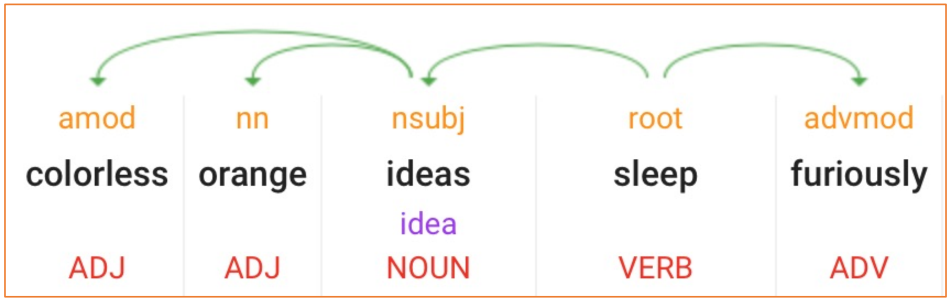
Example

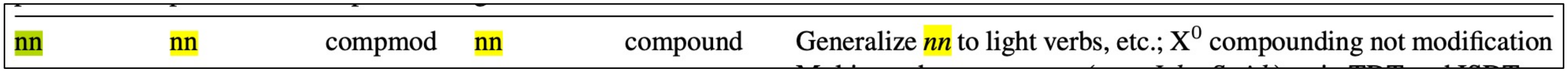
- Chomsky (p.c.): (11) is no longer an ideal example to use (*as it has been well-discussed over the last half century or so*), suggesting, instead, that substituting *orange* for *green* would reset the bigram statistics



11 *colorless green ideas sleep furiously*

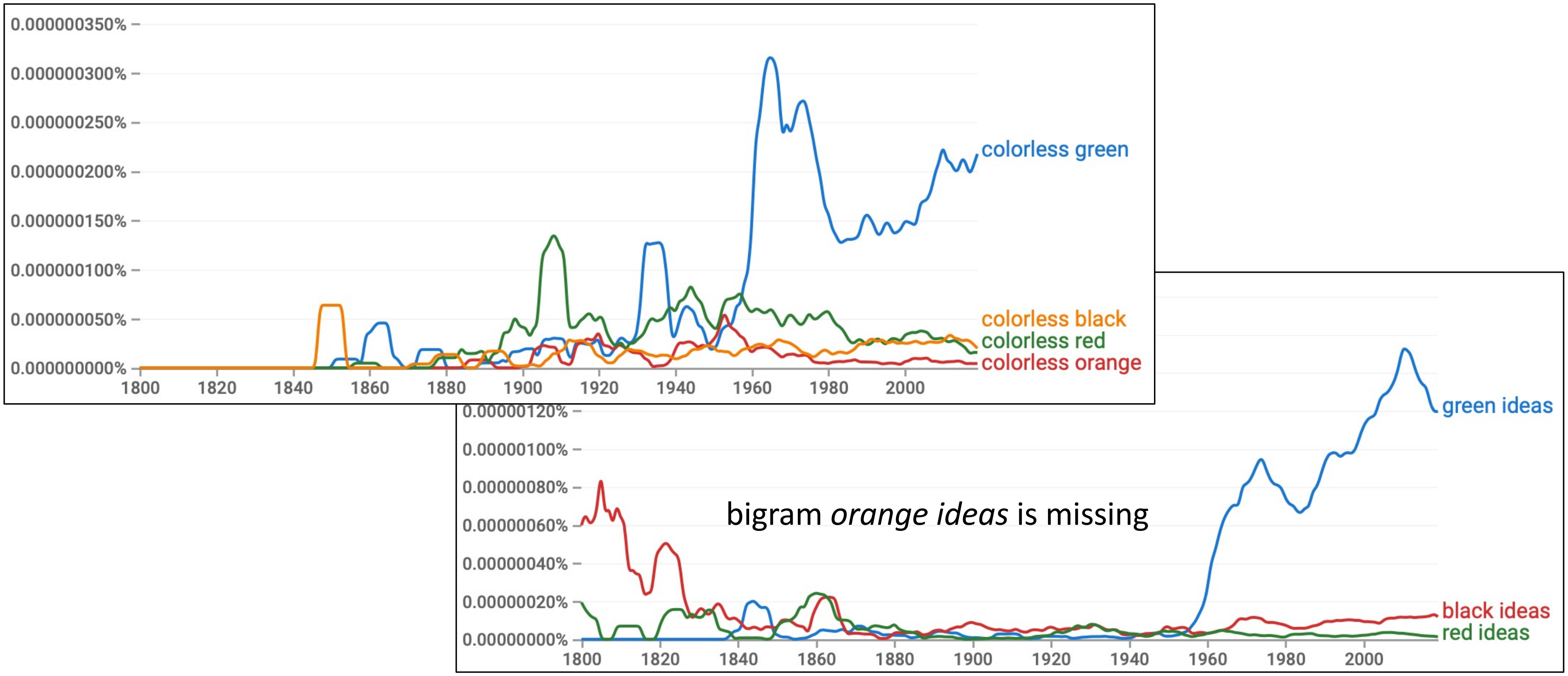
12 *furiously sleep ideas green colorless*





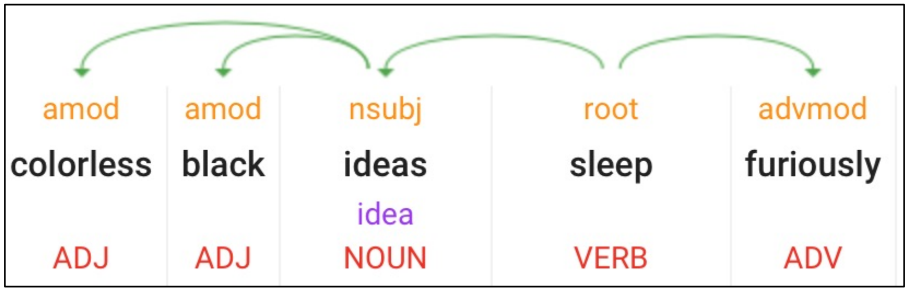
nn	nn	compmod	nn	compound	Generalize *nn* to light verbs, etc.; X^0 compounding not modification

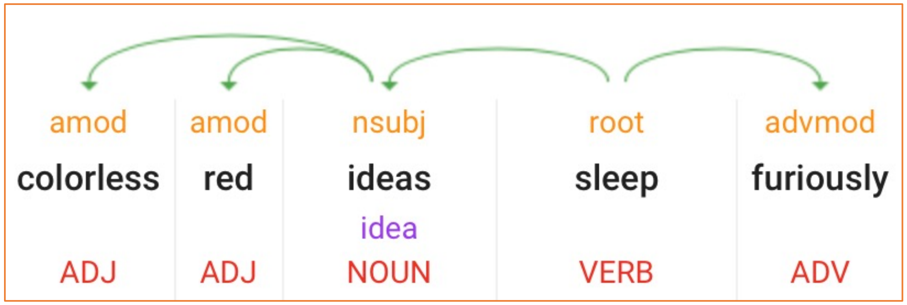
Example

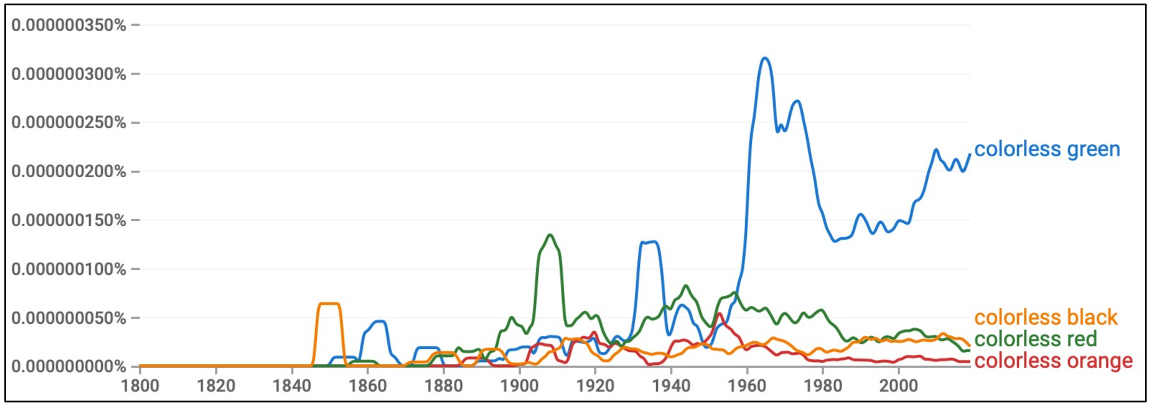


bigram *orange ideas* is missing

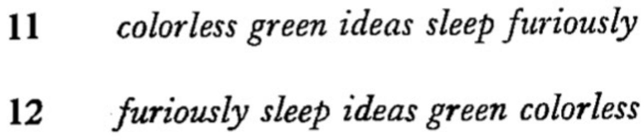
Example


amod
colorless
ADJ
amod
black
ADJ
nsubj
ideas
idea
NOUN
root
sleep
VERB
advmod
furiously
ADV


amod
colorless
ADJ
amod
red
ADJ
nsubj
ideas
idea
NOUN
root
sleep
VERB
advmod
furiously
ADV


0.000000350%
0.000000300%
0.000000250%
0.000000200%
0.000000150%
0.000000100%
0.000000050%
0.000000000%
1800
1820
1840
1860
1880
1900
1920
1940
1960
1980
2000
colorless green
colorless black
colorless red
colorless orange

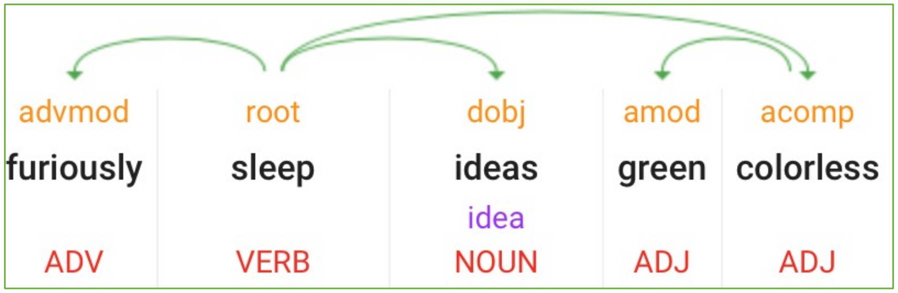
Example



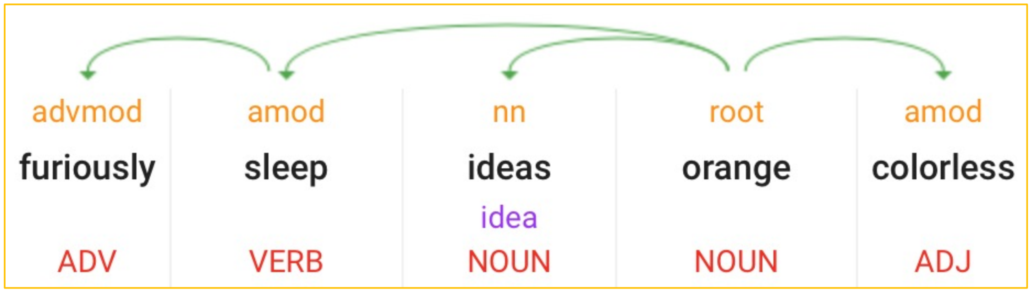
11 *colorless green ideas sleep furiously*

12 *furiously sleep ideas green colorless*

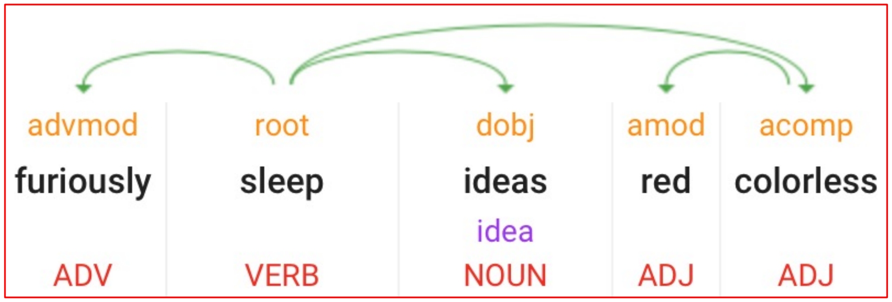
can you parse this?



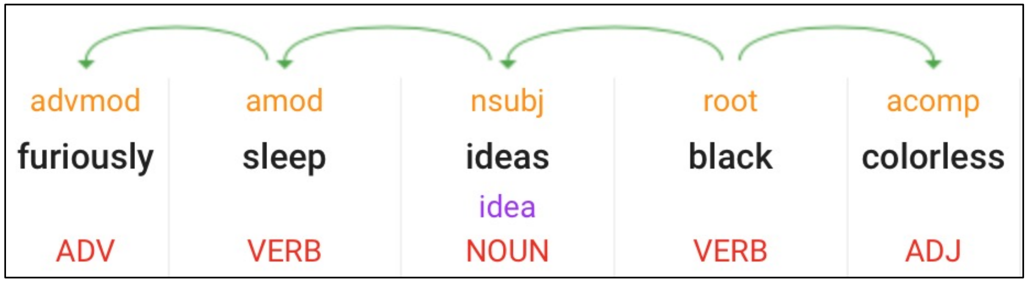
advmod	root	dobj	amod	acomp
furiously	sleep	ideas	green	colorless
		idea		
ADV	VERB	NOUN	ADJ	ADJ



advmod	amod	nn	root	amod
furiously	sleep	ideas	orange	colorless
		idea		
ADV	VERB	NOUN	NOUN	ADJ



advmod	root	dobj	amod	acomp
furiously	sleep	ideas	red	colorless
		idea		
ADV	VERB	NOUN	ADJ	ADJ



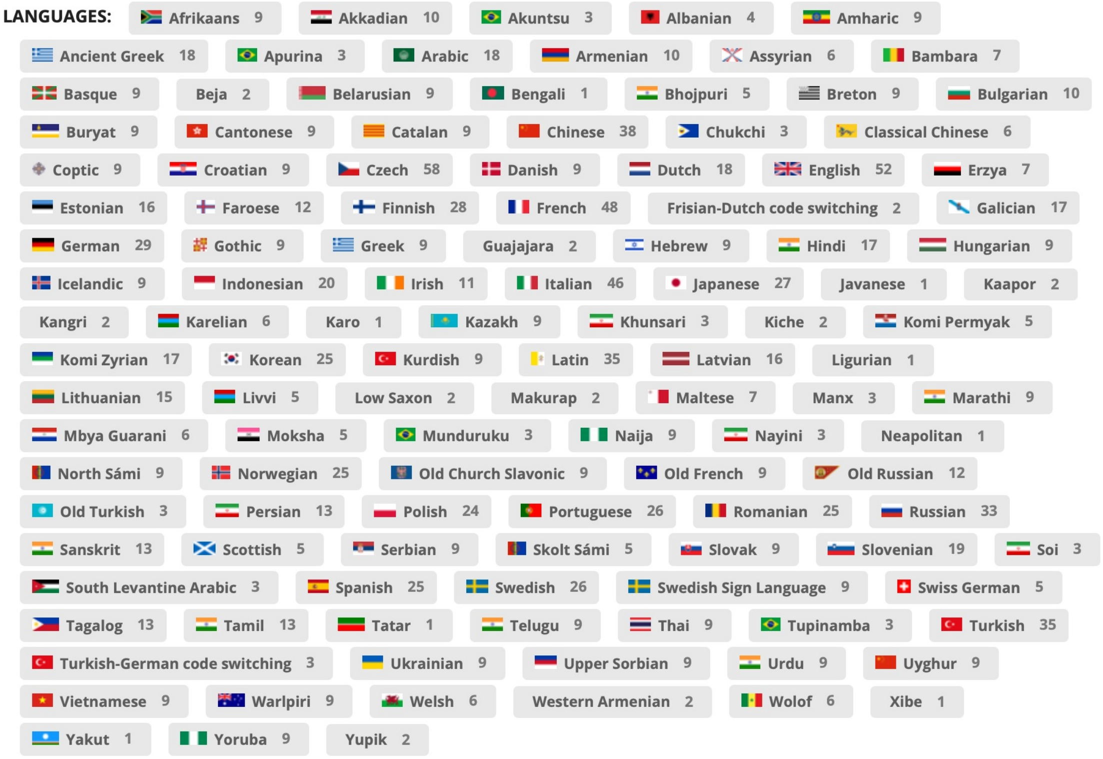
advmod	amod	nsubj	root	acomp
furiously	sleep	ideas	black	colorless
		idea		
ADV	VERB	NOUN	VERB	ADJ

Example

- We get 3 different sequences:
    1. ADV VERB NOUN ADJ ADJ
    2. ADV VERB NOUN NOUN ADJ
    3. ADV VERB NOUN VERB ADJ
- Question: *are these grammatical sequences*?
    - **Quick Homework 4** (*do it before next time*)
    - Come up with grammatical examples of these three sequences (in English), or explain why you think they're not.
    - Email to me (*usual rules*)

Universal Dependencies

- Publicly available dependency treebanks
- Produced by parsers or by human experts or some combination thereof?
- Lots of training data for machine learning



LANGUAGES:

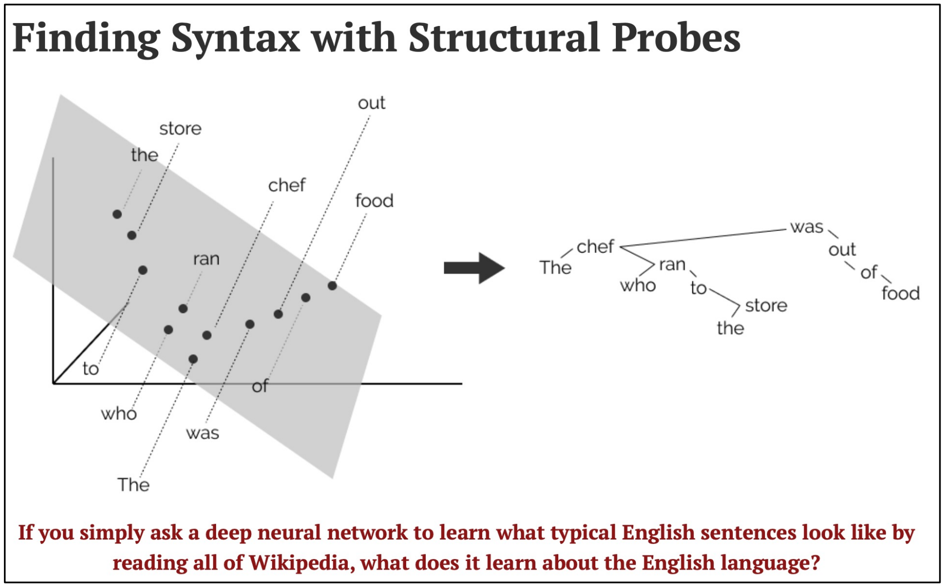
Afrikaans	9
Akkadian	10
Akuntsu	3
Albanian	4
Amharic	9
Ancient Greek	18
Apurina	3
Arabic	18
Armenian	10
Assyrian	6
Bambara	7
Basque	9
Beja	2
Belarusian	9
Bengali	1
Bhojpuri	5
Breton	9
Bulgarian	10
Buryat	9
Cantonese	9
Catalan	9
Chinese	38
Chukchi	3
Classical Chinese	6
Coptic	9
Croatian	9
Czech	58
Danish	9
Dutch	18
English	52
Erzya	7
Estonian	16
Faroese	12
Finnish	28
French	48
Frisian-Dutch code switching	2
Galician	17
German	29
Gothic	9
Greek	9
Guajajara	2
Hebrew	9
Hindi	17
Hungarian	9
Icelandic	9
Indonesian	20
Irish	11
Italian	46
Japanese	27
Javanese	1
Kaapor	2
Kangri	2
Karelian	6
Karo	1
Kazakh	9
Khunsari	3
Kiche	2
Komi Permyak	5
Komi Zyrian	17
Korean	25
Kurdish	9
Latin	35
Latvian	16
Ligurian	1
Lithuanian	15
Livvi	5
Low Saxon	2
Makurap	2
Maltese	7
Manx	3
Marathi	9
Mbya Guarani	6
Moksha	5
Munduruku	3
Naija	9
Nayini	3
Neapolitan	1
North Sámi	9
Norwegian	25
Old Church Slavonic	9
Old French	9
Old Russian	12
Old Turkish	3
Persian	13
Polish	24
Portuguese	26
Romanian	25
Russian	33
Sanskrit	13
Scottish	5
Serbian	9
Skolt Sámi	5
Slovak	9
Slovenian	19
Soi	3
South Levantine Arabic	3
Spanish	25
Swedish	26
Swedish Sign Language	9
Swiss German	5
Tagalog	13
Tamil	13
Tatar	1
Telugu	9
Thai	9
Tupinamba	3
Turkish	35
Turkish-German code switching	3
Ukrainian	9
Upper Sorbian	9
Urdu	9
Uyghur	9
Vietnamese	9
Warlpiri	9
Welsh	6
Western Armenian	2
Wolof	6
Xibe	1
Yakut	1
Yoruba	9
Yupik	2

Structural Probes

- Paper (Hewitt & Manning, 2019)
  - https://nlp.stanford.edu/pubs/hewitt2019structural.pdf
- Blog entry:
  - https://nlp.stanford.edu//~johnhew//structural-probe.html

- **Idea**:

1. use contextual word embeddings (ELMo or BERT)
2. No dependency treebank
3. "*We'll present a method for finding tree structures in these vector spaces, and show the surprising extent to which ELMo and BERT encode human-like parse trees.*"



Finding Syntax with Structural Probes

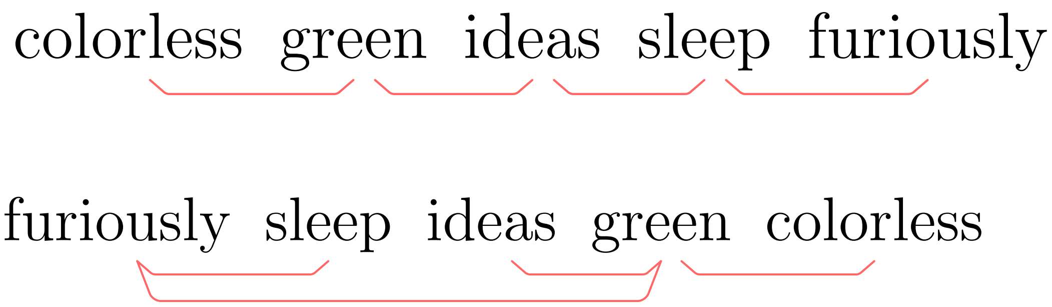
If you simply ask a deep neural network to learn what typical English sentences look like by reading all of Wikipedia, what does it learn about the English language?

Structural Probes

- You'll install this code on your computer
  - Python 3 required, plus some modules
  - https://github.com/john-hewitt/structural-probes
- We'll discuss the installation and Homework 5 next lecture.
- Example:

```
"The chef that went to the stores was out of food" | python structural-probes/run_demo.py
example/demo-bert.yaml
Constructing new results directory at example/results/BERT-disk-demo-2022-1-28-14-40-51-
39486/
The pre-trained model you are loading is a cased model but you have not set
`do_lower_case` to False. We are setting `do_lower_case=False` for you but you may want
to check this behavior.
100%|███████████████████████████████| 213450/213450 [00:00<00:00, 2492466.95B/s]
100%|███████████████████████| 1242874899/1242874899 [08:11<00:00, 2530392.20B/s]
Constructing TwoWordPSDProbe
Constructing OneWordPSDProbe
[demoing]: 1it [00:00,  1.26it/s]
```

Structural Probes



colorless green ideas sleep furiously

furiously sleep ideas green colorless