



LING 388: Computers and Language

Lecture 25

Today's Topics

- Homework 9 Review
- Homework 10: Term project proposals please!
 1. just one paragraph or page, email to me
 2. describe what you want to do
 3. should not be too small a project
 4. should not be too big
 5. must be related to something taught in class this semester
- Some final words about Masked Language Models
- Syntax: some parsers available online

Homework 9 Review

- Q1: Vector arithmetic
 - **Idea:** dogs – dog computes a NUM vector in direction PL to SG
 - Test it on irregular plural nouns: can it correctly pick the singular?

17/30 right, 13 wrong

Irregular plural nouns in English promova

Irregular plural nouns list

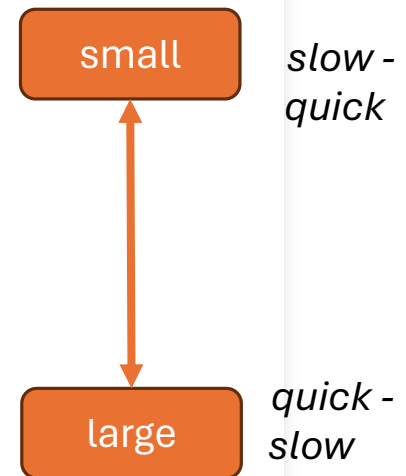
• Child/Children	• Foot/Feet	• Deer/Deer	• Analysis/Analyses	• Thesis/Theses
• Man/Men	• Tooth/Teeth	• Moose/Moose	• Basis/Bases	• Axis/Axes
• Woman/Women	• Fungus/Fungi	• Crisis/Crises	• Diagnosis/Diagnoses	• Appendix/Appendices
• Person/People	• Cactus/Cacti	• Wolf/Wolves	• Hypothesis/Hypotheses	• Nucleus/Nuclei
• Mouse/Mice	• Sheep/Sheep	• Index/Indices	• Oasis/Oases	• Focus/Foci
• Goose/Geese	• Fish/Fish	• Ox/Oxen	• Synthesis/Syntheses	• Matrix /Matrices

```
>>> import gensim.downloader
>>> model = gensim.downloader.load('glove-wiki-gigaword-50')
```

Homework 9 Review

Question 2: can you find **two good examples** of antonym pairs satisfying equations:

1. *slow* + *<polar adjective>* - *quick*, and
2. *quick* + *<antonym>* - *slow* = *<polar adjective>*



Homework 9 Review

Question 2: many don't work

- e.g. *white* <-> *black* works but not *tiny* or *huge*

```
>>> ['{} {:.2f}'.format(w,p) for (w,p) in
model.most_similar(positive=['slow','white'],negative=['quick'],topn=3)]
['black 0.79', 'gray 0.78', 'covered 0.74']
```

```
>>> ['{} {:.2f}'.format(w,p) for (w,p) in
model.most_similar(positive=['quick','black'],negative=['slow'],topn=3)]
['white 0.81', 'red 0.73', 'green 0.73']
```

```
>>> ['{} {:.2f}'.format(w,p) for (w,p) in
model.most_similar(positive=['slow','tiny'],negative=['quick'],topn=3)]
['small 0.73', 'dense 0.72', 'parts 0.72']
```

```
>>> ['{} {:.2f}'.format(w,p) for (w,p) in
model.most_similar(positive=['slow','huge'],negative=['quick'],topn=3)]
['massive 0.79', 'enormous 0.76', 'partly 0.76']
```

Homework 9 review

- Plenty of pairs if you do a search ...

Table 3. Polar adjective pairs used in AttrakDiff questionnaire.

technical – human	complicated – simple	impractical – practical
cumbersome – straightforward	unpredictable – predictable	confusing – clearly structured
unruly – manageable	isolating – connective	unprofessional – professional
tacky – stylish	cheap – premium	alienating – integrating
separates me from people – brings me closer to people	not presentable – presentable	conventional – inventive
unimaginative – creative	cautious – bold	conservative – innovative
dull – captivating	undemanding – challenging	ordinary – novel
unpleasant – pleasant	ugly – attractive	disagreeable – likeable
rejecting – inviting	bad – good	repelling – appealing
discouraging – motivating		

MLM

Masked Language Modeling (MLM):

```
>>> from transformers import pipeline  
>>> classifier_large = pipeline("fill-mask", model='roberta-large')
```

MLM: significance of the period (.)

- `classifier_large('Noam Chomsky is a <mask>', top_k=7)]`
 1. ['Noam Chomsky is a liar
0.09',
 2. 'Noam Chomsky is a genius
0.06',
 3. 'Noam Chomsky is a fraud
0.05',
 4. 'Noam Chomsky is a traitor
0.02',
 5. 'Noam Chomsky is a hero
0.02',
 6. 'Noam Chomsky is a psychopath
0.02',
 7. 'Noam Chomsky is a Nazi
0.02']
- `classifier_large('Noam Chomsky is a <mask>.', top_k=7)]`
 1. ['Noam Chomsky is a genius.
0.09',
 2. 'Noam Chomsky is a radical.
0.04',
 3. 'Noam Chomsky is a
whistleblower. 0.04',
 4. 'Noam Chomsky is a visionary.
0.04',
 5. 'Noam Chomsky is a prophet.
0.03',
 6. 'Noam Chomsky is a liar.
0.03',
 7. 'Noam Chomsky is a
revolutionary. 0.03']

Constituency Parsing

- Requires a grammar (of English) – that's the **knowledge of language**
- **Parser** = a program that *finds* a parse using the grammar
- **Constituents** are phrases: e.g. NP, VP, PP and S.
- Finding a **parse** involves (proper) algorithms for dealing with grammars
 - otherwise, no guarantee of finding a parse (if one exists)
 - if one exists, return that **structural description**, i.e. parse.
 - if **structurally ambiguous**, proper response is to return all parses
 - if sentence is **ungrammatical**, proper response is **no parse**
 - i.e. detect that the sentence is bad,
 - and **not** just emit some random, incorrect parse

Looking ahead: writing a grammar

- Using nltk:

```
$ python
```

```
>>> import nltk
```

```
>>> g = open('g.txt').read()
```

```
>>> g
```

```
"s -> y x3\ns -> x1 z\ns -> y z\ny -> x1 x2\nz -> x2 x3\nx1 -> 'w1'\nx2 -> 'w2'\nx3 -> 'w3'\n"
```

```
>>> cfg = nltk.CFG.fromstring(g)
```

```
>>> cfg
```

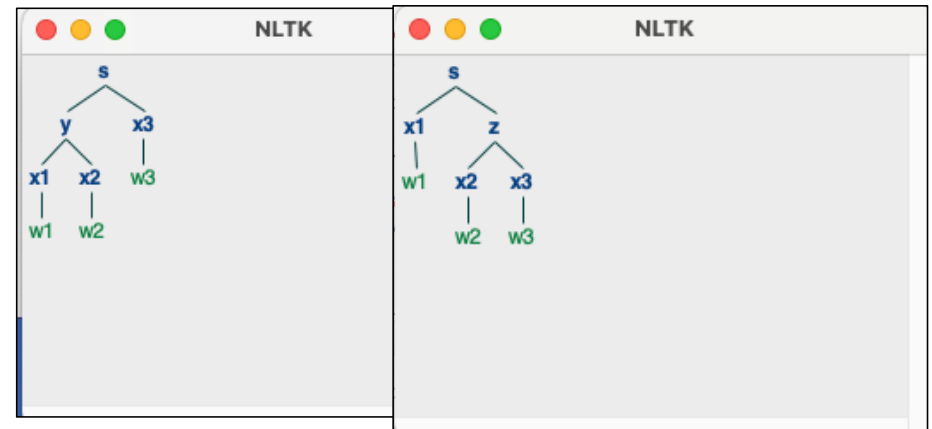
```
<Grammar with 8 productions>
```

```
>>> p = nltk.ChartParser(cfg)
```

```
>>> for tree in p.parse(['w1', 'w2', 'w3']):
```

```
...     tree.draw()
```

```
...
```



Parse could be a Picture

Filter: DEFAULT ▾


<https://creator.nightcafe.studio>

Publish 🗑️ 📄 ⌛ ➦ ⋮

A man saw the boy with a ball with a telescope.
Created a few seconds ago

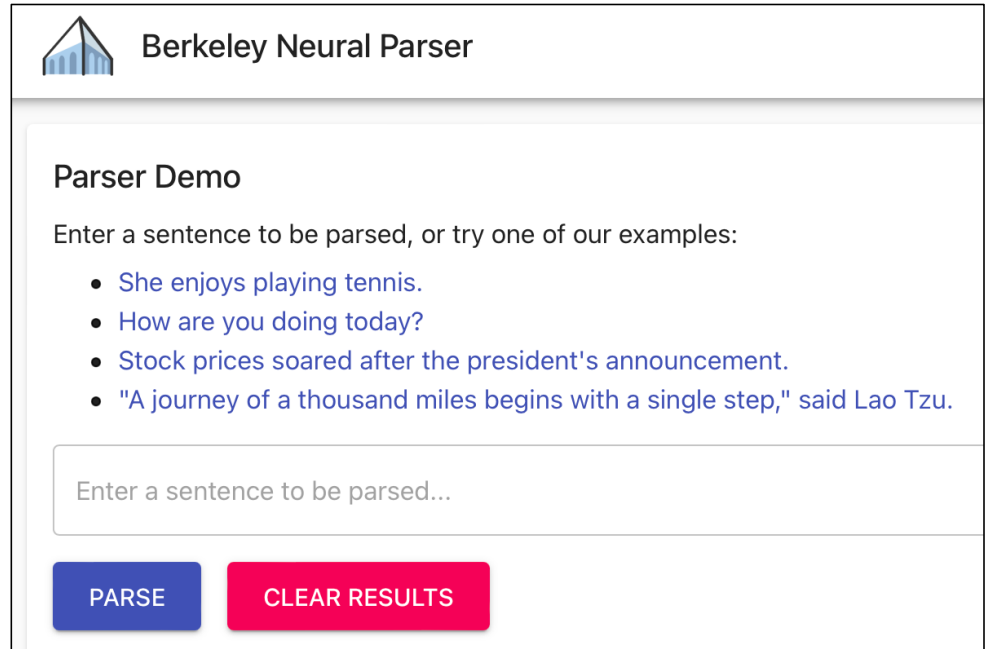
🗨️ Stable ↻ Short 🖼️ Thumb

EVOLVE THIS CREATION



Constituency Parsing

- Parsers are available online (trained on large corpora).
- Stanford CoreNLP
- Stanford Stanza
- Example:
 - Berkeley Neural Parser
 - <https://parser.kitaev.io>



The screenshot shows the Berkeley Neural Parser web interface. At the top left is a logo of a blue and white striped triangle. To its right is the text "Berkeley Neural Parser". Below this is a section titled "Parser Demo". Under the title, it says "Enter a sentence to be parsed, or try one of our examples:". There are four bullet points of example sentences: "She enjoys playing tennis.", "How are you doing today?", "Stock prices soared after the president's announcement.", and "A journey of a thousand miles begins with a single step," said Lao Tzu. Below the examples is a text input field with the placeholder text "Enter a sentence to be parsed...". At the bottom of the interface are two buttons: a blue button labeled "PARSE" and a red button labeled "CLEAR RESULTS".

Constituency Parsing

About

The Berkeley Neural Parser annotates a sentence with its syntactic structure by decomposing it into nested sub-phrases. See our [GitHub project](#) for information on how to install a standalone version of the parser and download models for 10+ languages, including English and Chinese. As of January 2019, our parser and models are state-of-the-art for all languages that we evaluate on.

This demo runs the version of the parser described in [Multilingual Constituency Parsing with Self-Attention and Pre-Training](#). The model used in the demo (benepar_en2) incorporates [BERT](#) word representations and achieves 95.17 F1 on the Penn Treebank.

Notice: it uses BERT embeddings
claim: the best there is!

References:

- [A Minimal Span-Based Neural Constituency Parser](#). Mitchell Stern, Jacob Andreas, Dan Klein. ACL 2017.
- [Constituency Parsing with a Self-Attentive Encoder](#). Nikita Kitaev and Dan Klein. ACL 2018.
- [Multilingual Constituency Parsing with Self-Attention and Pre-Training](#). Nikita Kitaev, Steven Cao, Dan Klein. ACL 2019.

Constituency Parsing

<http://stanza.run>

Stanza 1.6.0 (updated October 2023)

— Text to annotate —
e.g., The quick brown fox jumped over the lazy dog.

— Annotations —
parts-of-speech x named entities x lemmas x dependency parse x constituency parse x

— Language —
English

Submit

<https://corenlp.run>

 CoreNLP
version 4.5.5

— Text to annotate —
e.g., The quick brown fox jumped over the lazy dog.

— Annotations —
parts-of-speech x named entities x |

— Language —
English

Submit

- parts-of-speech
- lemmas
- named entities
- named entities (regexner)
- constituency parse**
- dependency parse

Constituency Parsing

- Let's try it:
 1. John saw the man
 2. John saw the man with a telescope
 3. *John saw (no object)
 4. *saw the man (no subject)
 5. *John saw man the (determiner-noun order reversed)
 6. *John the man saw (*like a head-final language*)
 7. *saw John the man (*like a VSO language*)

Constituency Parsing

- **Yoda speak** involves displacement:
 - *(still comprehensible to the native English speaker)*
 - “Found someone, you have, I would say, hmmm?”
 - “Much to learn, you still have.”
 - “Truly wonderful, the mind of a child is.”
 - “Lost a planet, Master Obi-Wan has.”
 - “That group back there, soon discovered will be.”
- Examples above from "An Unusual Way of Speaking, Yoda Has" by Adrienne LaFrance in *The Atlantic* (Dec 2015).
 - <https://www.theatlantic.com/entertainment/archive/2015/12/hmmmmm/420798/>

