

LING/C SC/PSYC 438/538

Lecture 2

Sandiway Fong

Administrivia

Do you have Perl and Python3 installed on your computer?

- That was Homework 2
- Please try to have it ready by next week!

Today's Lecture

Chapter 1 of JM Reading:

- That was Homework 1. A brief quiz today. End of lecture.

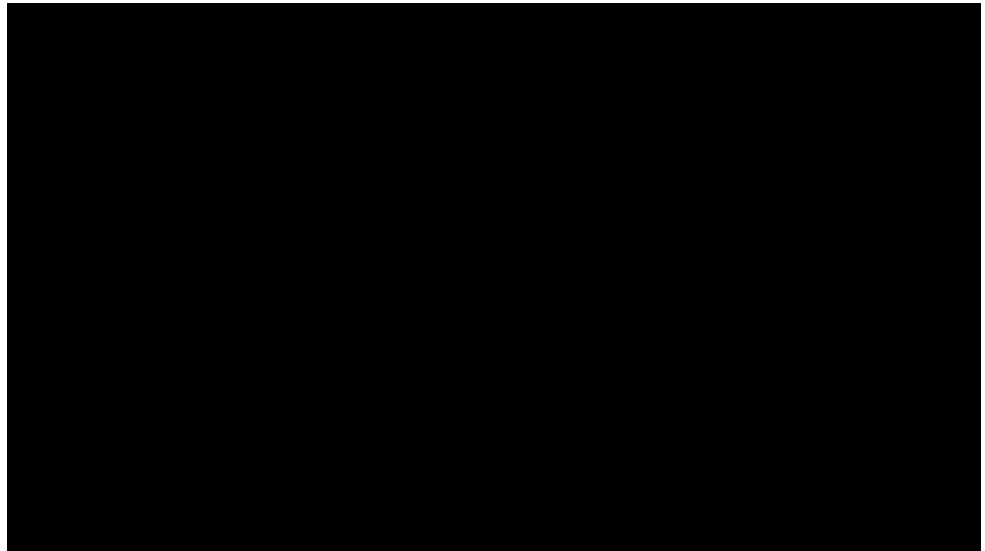
Introduction

- Sentence analysis

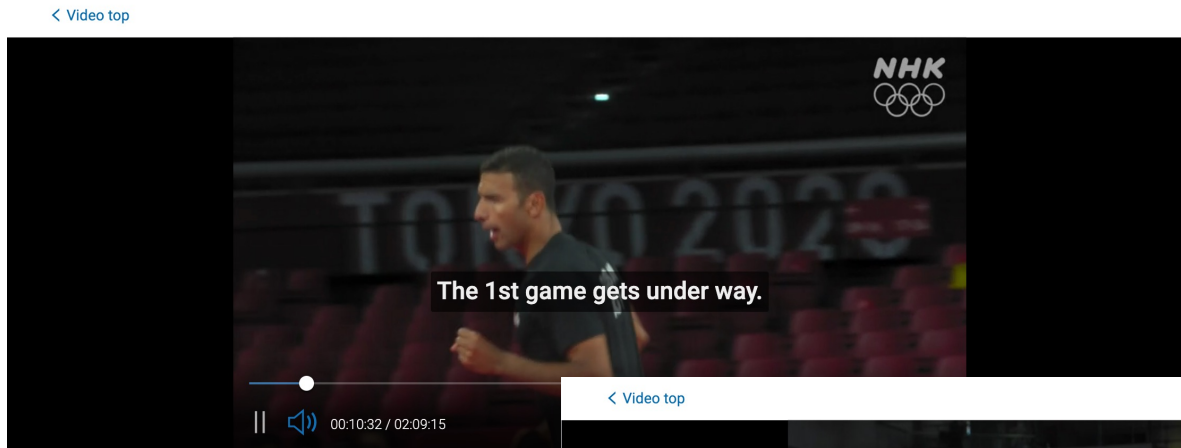
Language and Computers

- We'd like computers to be smart about language
 - *there's plenty of training data around*
 - feed AI/machine learning, make the machines learn by themselves
 - pass the **Turing Test**, but not be too smart?

HAL 9000 in *2001: A Space Odyssey*



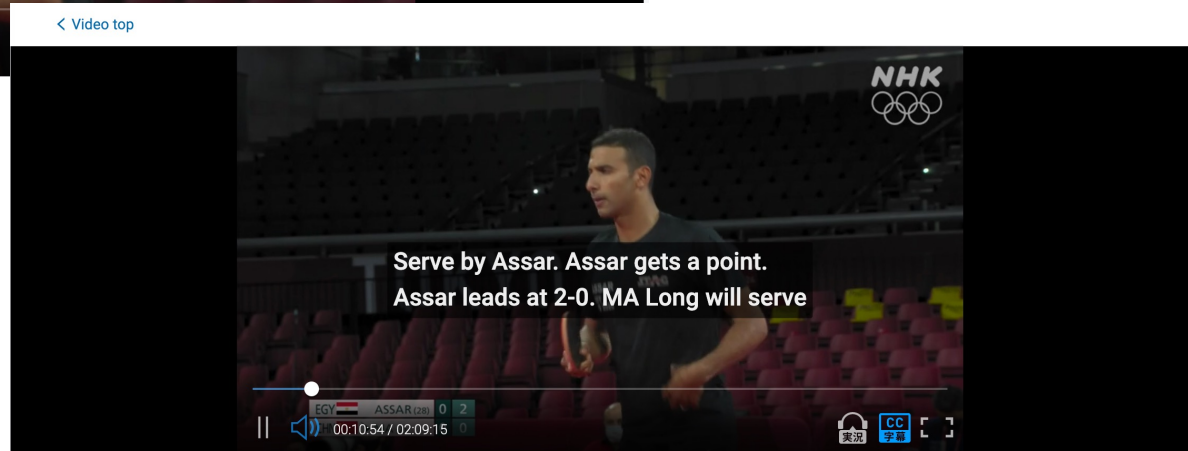
Assistive Technology: point-by-point subtitles



The subtitles in this stream are brought by "robot play-by-play broadcast". It is automatically generated by computers using data provided by the organizer of the event. The contents may differ from the actual live voice coverage.



← subtitle



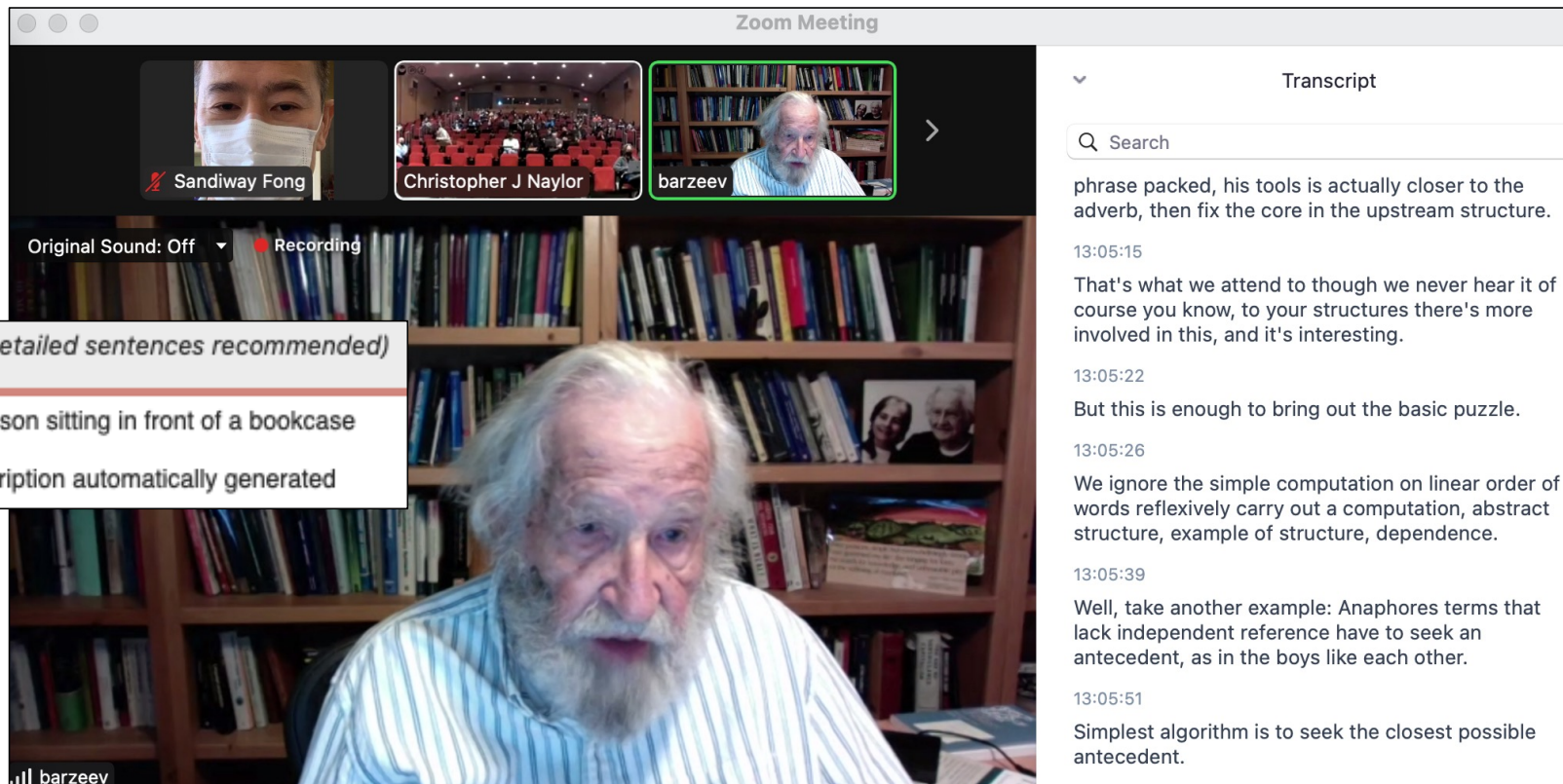
Assistive Technology: Zoom transcript

Alt Text

(1-2 detailed sentences recommended)

A person sitting in front of a bookcase

Description automatically generated



Zoom Meeting

Sandiway Fong

Christopher J Naylor

barzeev

Original Sound: Off

Recording

Transcript

Search

phrase packed, his tools is actually closer to the adverb, then fix the core in the upstream structure.

13:05:15

That's what we attend to though we never hear it of course you know, to your structures there's more involved in this, and it's interesting.

13:05:22

But this is enough to bring out the basic puzzle.

13:05:26

We ignore the simple computation on linear order of words reflexively carry out a computation, abstract structure, example of structure, dependence.

13:05:39

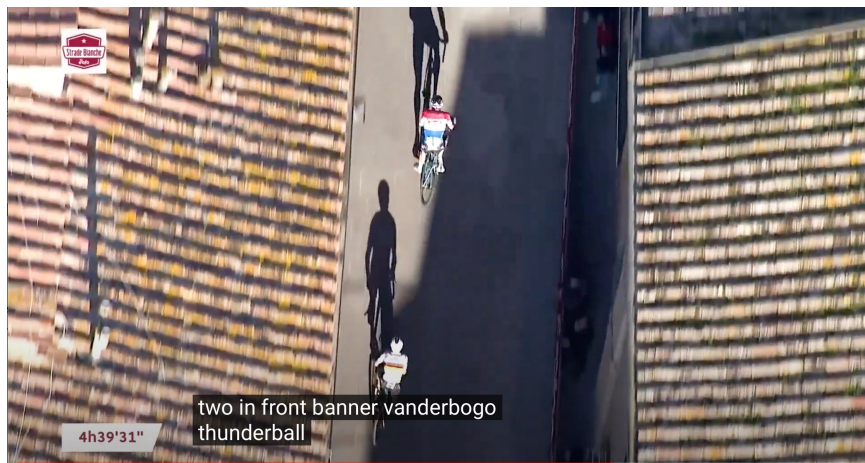
Well, take another example: Anaphores terms that lack independent reference have to seek an antecedent, as in the boys like each other.

13:05:51

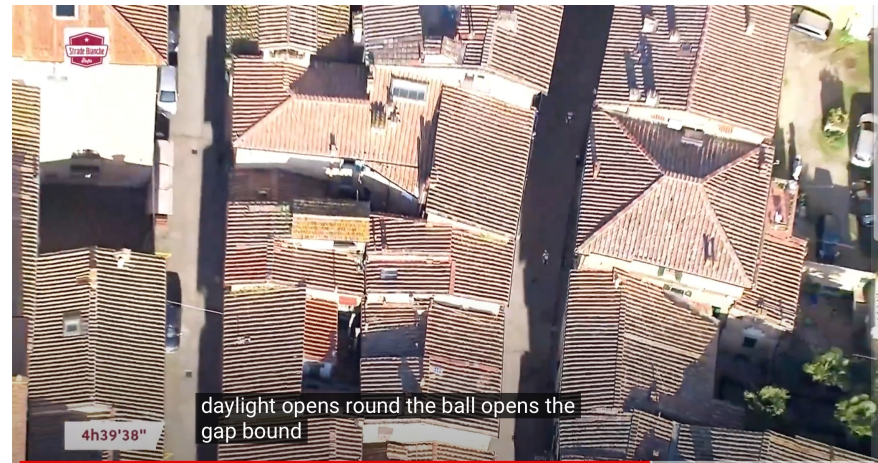
Simplest algorithm is to seek the closest possible antecedent.

barzeev

Harder task: names (CC)



- banner vanderbogo



- round the ball

Harder task: names (CC)

Mathieu van der Poel



- matthew vanderbald



- vanderbolt

Language and Computers

- (Un)fortunately, we're not quite there yet...
 - *still a gap between what computers can do and what we want them to be able to do*

Often quoted (**but never verified**):

"The spirit is strong, but the flesh is weak" was translated into [Russian and then back to English, the result was "The vodka is good, but the meat is rotten."](#)

but with Google translate or ~~babelfish~~, it's not difficult to find (funny) examples...

Language and Computers

- and how can we tell if the translation is right anyway?



<http://fun.drno.de/pics/english/only-in-china/TranslateServerError.jpg>

Applications

- ChatGPT dialog sentences are natural and grammatical
 - *a pretty big achievement IMHO*
- but even if we are willing to pay...
 - machine translation has been worked on since after World War II
 - *still not perfected today*
 - **why?**
 - what are the properties of human languages that make it hard?

Language and Computers

We can exploit the recursive nature of language ...

Biden apologizes to Obama for marriage controversy

From **Jessica Yellin**, CNN Chief White House Correspondent
updated 10:20 PM EDT, Thu May 10, 2012

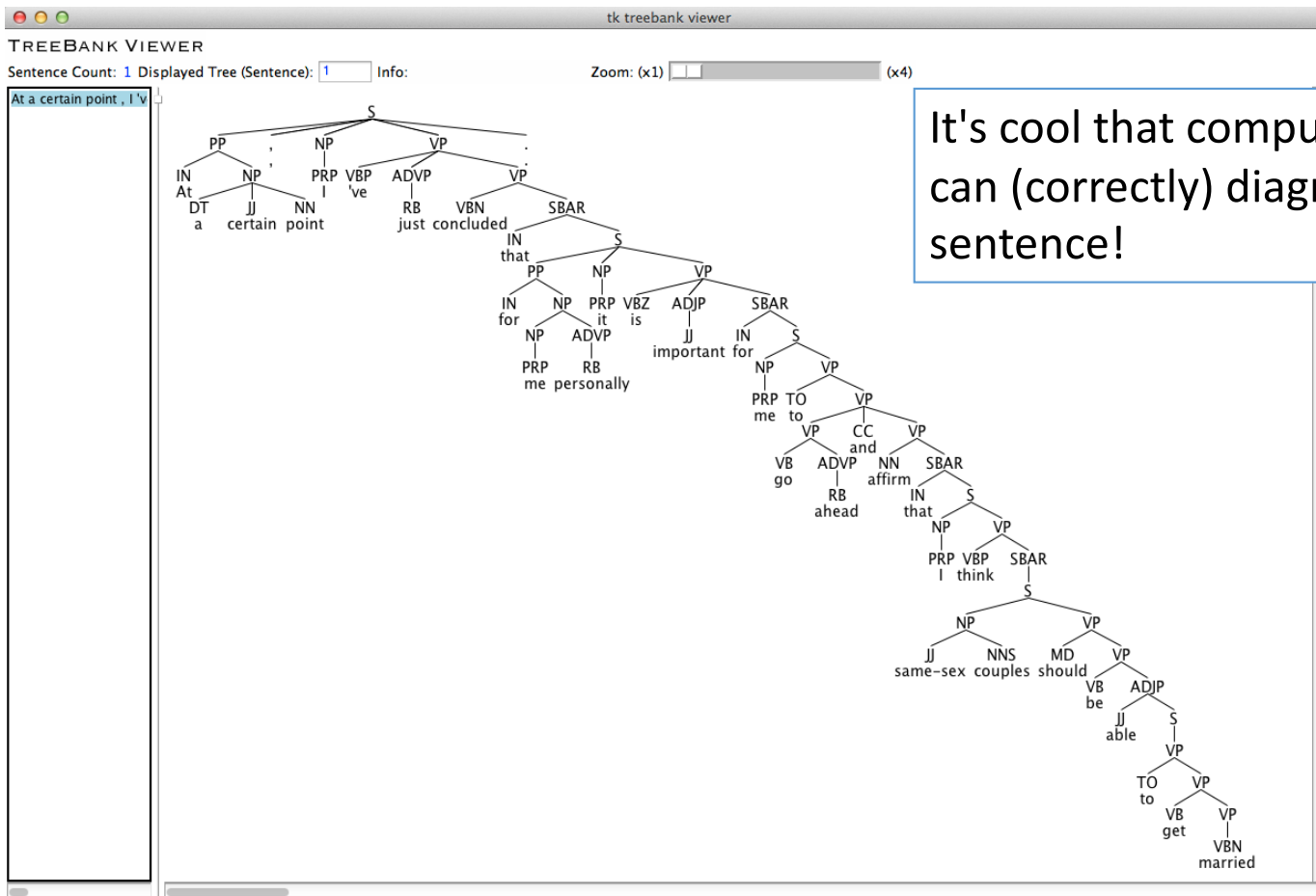


Language and Computers

- Obama: *"At a certain point, I've just concluded that for me personally it is important for me to go ahead and affirm that I think same-sex couples should be able to get married."*

Is this sentence complicated? Why?

Language and Computers



It's cool that computer programs can (correctly) diagram this long sentence!

Language and Computers

May 9, 2012 2:57pm

Obama: 'I Think Same-Sex Couples Should Be Able to Get Married'

Like 60k Tweet 845 +1 80

178 Text



Executive
Summarization

Language and Computers

- Obama: "~~At a certain point, I've just concluded that for me personally it is important for me to go ahead and affirm that~~ **I think same-sex couples should be able to get married.**"

Most summarizer programs can't do this ...

Language and Computers

A Sports Shooter Shoots Shooters Shooting Sports

Jul 12, 2015 · Michael Zhang

[Share](#) [Like](#) 1.2k

15 Comments



Language and Computers

Natural language parsers

- Let's get some idea of what these (popular) systems produce.

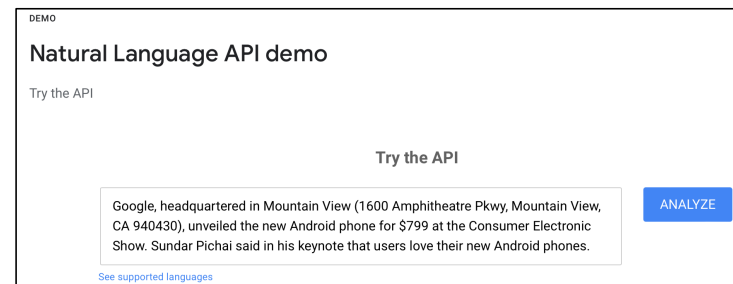
Sadly, all once had an easily accessible working demo page:

1. Stanford Parser
2. Berkeley Parser
3. Google Natural Language

(**deprecated?** Try <https://corenlp.run>)

(**deprecated?** B. Neural Parser: <https://parser.kitaev.io>)

(<https://cloud.google.com/natural-language>)



Language and Computers



The Stanford Natural Language Processing Group

[people](#)

[publications](#)

[research blog](#)

[software](#)

[teaching](#)

[join](#)

[local](#)

Software > Stanford Parser

[About](#) | [Citing](#) | [Questions](#) | [Download](#) | [Included Tools](#) | [Extensions](#) | [Release history](#) | [Sample output](#) | [Online](#) | [FAQ](#)

About

A natural language parser is a program that works out the grammatical **structure of sentences**, for instance, which groups of words go together (as "phrases") and which words are the **subject** or **object** of a verb. Probabilistic parsers use knowledge of language gained from hand-parsed sentences to try to produce the *most likely* analysis of new sentences. These statistical parsers still make some mistakes, but commonly work rather well. Their development was one of the biggest breakthroughs in natural language processing in the 1990s. You can [try out our parser online](#).

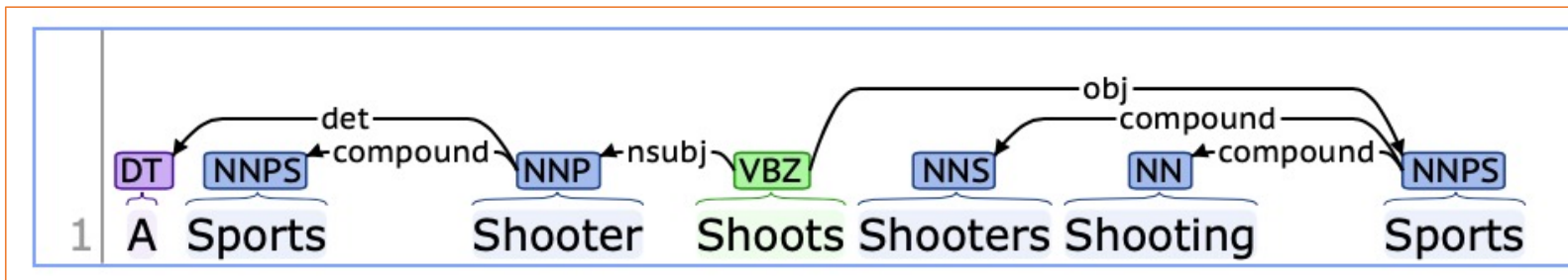
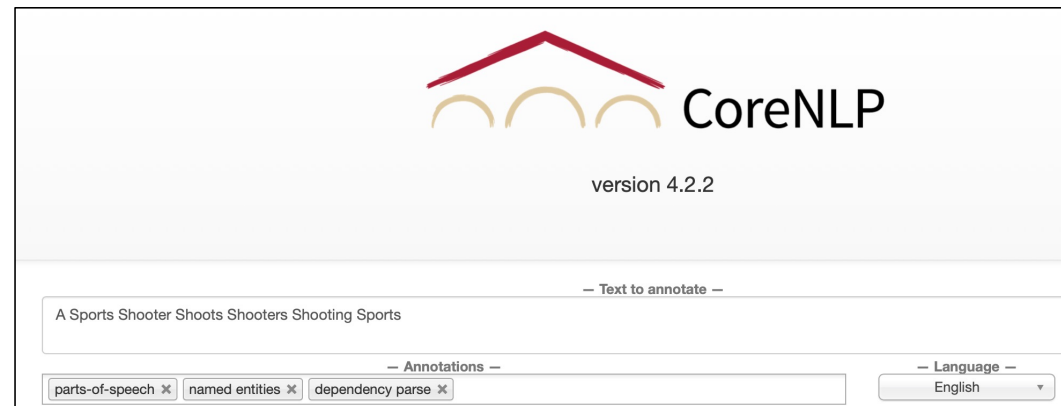
<http://nlp.stanford.edu:8080/parser/>

You can download the code.
Server no longer up and running.

Language and Computers

- Natural language parsers
 - Stanford CoreNLP
 - Java-based
 - Demo!

(<https://corenlp.run>)



Language and Computers

- Natural language parsers
 - (Stanford) Stanza
 - Python-based (CoreNLP)
 - Demo!

<http://stanza.run>

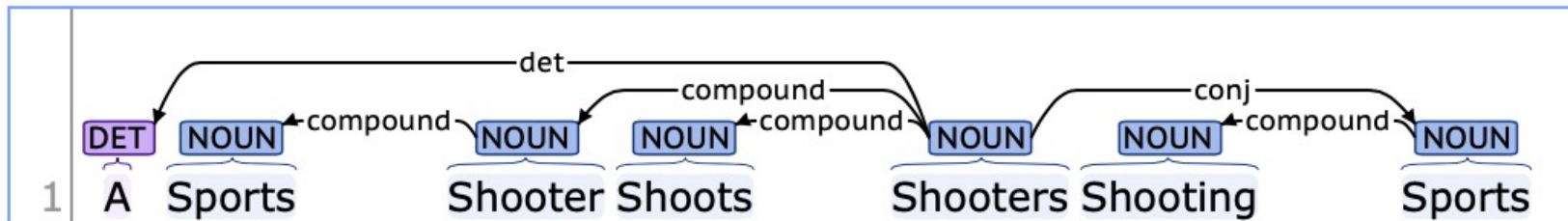
– Text to annotate –

A Sports Shooter Shoots Shooters Shooting Sports

– Annotations –

parts-of-speech × named entities × lemmas × dependency parse ×

Universal Dependencies:



Language and Computers

- Stanford parser <http://nlp.stanford.edu:8080/parser/index.jsp>

Stanford Parser

Please enter a sentence to be parsed:

A Sports Shooter Shoots Shooters Shooting Sports

- Part of Speech Tagging:

Tagging

A/DT Sports/NNPS Shooter/NNP Shoots/NNP Shooters/NNP Shooting/NNP Sports/NNP

DT = determiner; NNP = Proper Noun; NNPS = Plural Proper Noun;
VBZ = Verb 3rd Person Singular Present; VBG = Verb Gerund Form

Language and Computers

- Syntax (Constituency-based):

Parse

```
(ROOT
  (FRAG
    (NP (DT A) (NNPS Sports))
    (NP (NNP Shooter) (NNP Shoots) (NNP Shooters) (NNP Shooting) (NNP Sports))))
```

Constituents:

FRAG = Fragment (of a sentence)

S = Sentence, NP = Noun Phrase, VP = Verb Phrase

Parts of Speech:

DT Determiner, NNP Proper Noun, NNPS Plural Proper Noun.

Language and Computers

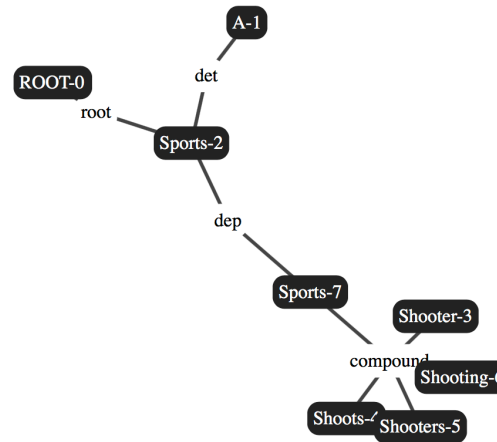
- Syntax (Dependencies):

Universal dependencies, enhanced

```
det(Sports-2, A-1)
root(ROOT-0, Sports-2)
compound(Sports-7, Shooter-3)
compound(Sports-7, Shoots-4)
compound(Sports-7, Shooters-5)
compound(Sports-7, Shooting-6)
dep(Sports-2, Sports-7)
```

RELATION(depends-on, dependent)

Typed Dependency Graph



Language and Computers

Your query

A sports shooter shoots shooters shooting sports

Tagging

A/DT sports/NNS shooter/NN shoots/VBZ shooters/NNS shooting/VBG sports/NNS

Parse

```
(ROOT
 (S
  (NP (DT A) (NNS sports) (NN shooter))
  (VP (VBZ shoots)
    (NP
     (NP (NNS shooters))
     (VP (VBG shooting)
        (NP (NNS sports)))))))))
```

FRAG = Fragment (of a sentence)
S = Sentence
NP = Noun Phrase
VP = Verb Phrase

Input: Choose File no file selected

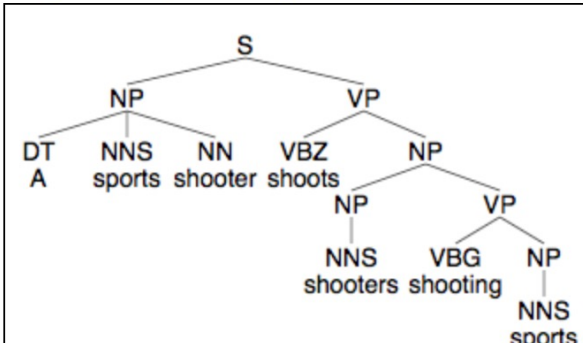
(VP (VBZ shoots) (NP (NP (NNS shooters))

WebSocket Status: 9... CONNECTED DISCONNECTED

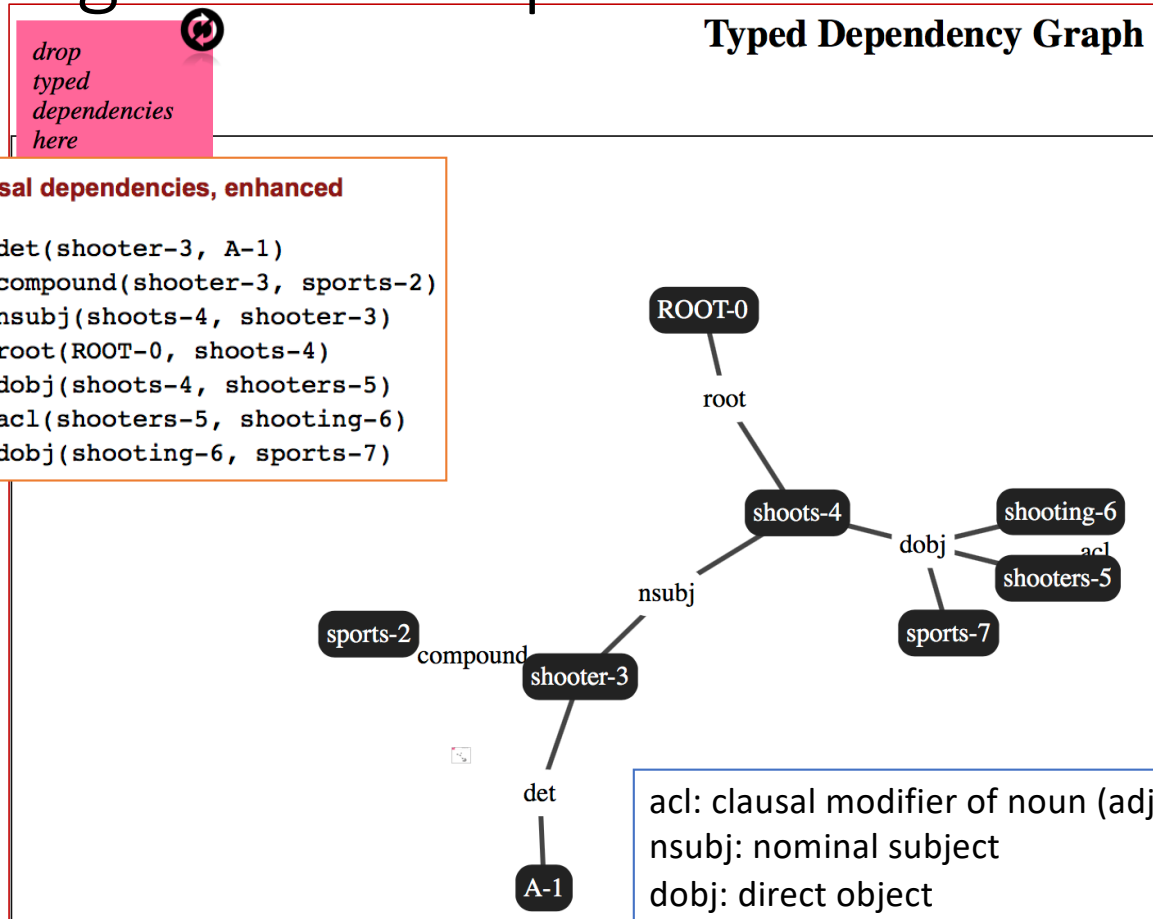
Help: (Typing pid↵ into the input box displays the process I

Four formats are accepted (however, they may not be mi

1: [expand](#) 2: [expand](#) 3: [expand](#) 4: [expand](#)



Language and Computers



Language and Computers

- Berkeley Parser (downloadable, *but no longer available online*)

<http://tomato.banatao.berkeley.edu:8080/parser/parser.html>

```
(ROOT
 (NP (DT A) (NNP Sport) (NNP Shooter) (NNP Shoots) (NNP Shooters) (NNP Shooting) (NNP Sports)))
```

```
      ROOT
      |
      NP
     /| | | | | |
    DT NNP NNP NNP NNP NNP NNP
    |  |  |  |  |  |  |
    A Sport Shooter Shoots Shooters Shooting Sports
```

A Sport Shooter Shoots Shooters Shooting Sports

Parse!

Language and Computers

Berkeley Parser

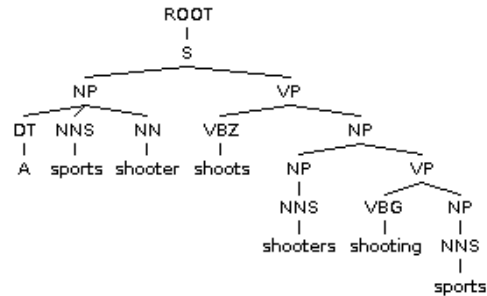
<http://tomato.banatao.berkeley.edu:8080/parser/parser.html>

This one is available

Berkeley Neural Parser

<https://parser.kitaev.io>

```
(ROOT
 (S
  (NP (DT A) (NNS sports) (NN shooter))
  (VP (VBZ shoots)
    (NP
     (NP (NNS shooters))
     (VP (VBG shooting)
        (NP (NNS sports)))))))
```



FRAG = Fragment (of a sentence)
S = Sentence
NP = Noun Phrase
VP = Verb Phrase

A sports shooter shoots shooters shooting sports

Parse!

Language and Computers

Universal Dependency Relations

The following table lists the 37 universal syntactic relations used in UD v2. It is a revised version of the relations originally described in *Universal Stanford Dependencies: A cross-linguistic typology* (de Marneffe et al. 2014).

The upper part of the table follows the main organizing principles of the UD taxonomy such that *rows* correspond to functional categories in relation to the head (core arguments of clausal predicates, non-core dependents of clausal predicates, and dependents of nominals) while *columns* correspond to structural categories of the dependent (nominals, clauses, modifier words, function words). The lower part of the table lists relations that are not dependency relations in the narrow sense.

	Nominals	Clauses	Modifier words	Function Words
Core arguments	nsubj obj iobj	csubj ccomp xcomp		
Non-core dependents	obl vocative expl dislocated	advcl	advmod* discourse	aux cop mark
Nominal dependents	nmod appos nummod	acl	amod	det clf case
Coordination	MWE	Loose	Special	Other
conj cc	fixed flat compound	list parataxis	orphan goeswith reparandum	punct root dep

* The *advmod* relation is used for modifiers not only of predicates but also of other modifier words.

<https://universaldependencies.org/u/dep/index.html>

Homework 3: Quick Quiz Questions

- (Knowledge of) **Semantics** — knowledge of meaning
 1. What is the difference between *a picture of John* and *a picture of John's*?
 2. Which is right? A) *a friend of John*, B) *a friend of John's*, or C) both
 3. Which is right? A) *a friend of mine*, B) *a friend of me*, or C) both
 4. Which is right? A) *a friend of hers*, B) *a friend of her*, or C) both
 5. Which is right? A) *glass of water*, B) *water's glass*, C) both
 6. What is the difference between *at the water's edge* and *at the edge of water*? Or no difference.
- We say some input is **ambiguous** if multiple, alternative linguistic structures can be built for it.
 7. Is *Caesar's murder* ambiguous? Explain.

Homework 3

- Email to sandiway@arizona.edu
- By Friday midnight
- SUBJECT: 438/538 Homework 3: YOUR NAME
- Either Plain Text or PDF accepted (no Word files please)