Linguistics in computational linguistics (better with Python)

Olga Zamaraeva

Universidade da Coruña
CITIC
For PyBirras

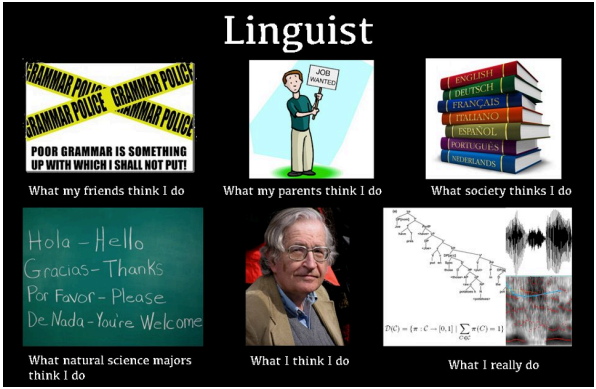
January 20 2024

Disclaimer

- This is not about Large Language Models

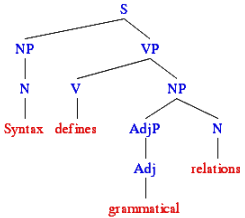
Introduction

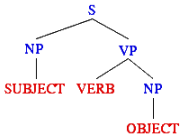
- I do academic research in linguistics
  - (not a Python expert)
- I will talk about why computational side matters in my research
- I'll briefly present a Python library that makes my research faster
- NB: Python is the lingua franca of academia

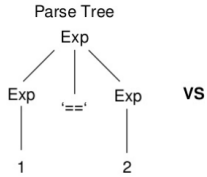


Also, don't check this out: https://xkcd.com/114/

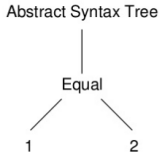
What is syntax?



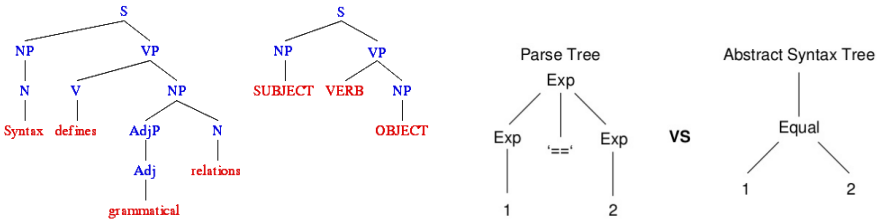




VS

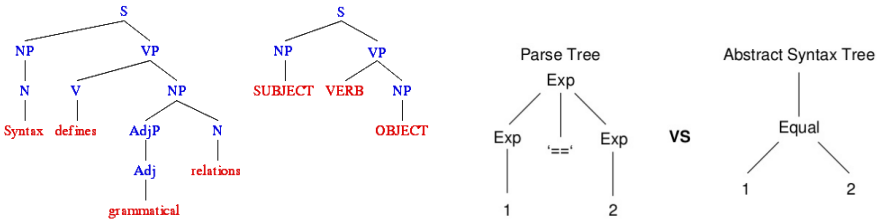


What is syntax?



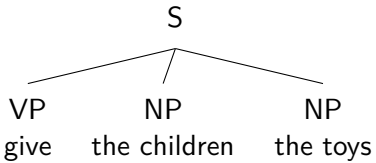
- A theory behind structure of phrase as scaffolding for the sentence meaning
- ...in linguistics
- ...in programming languages
- Main difference?

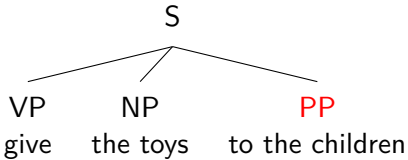
What is syntax?

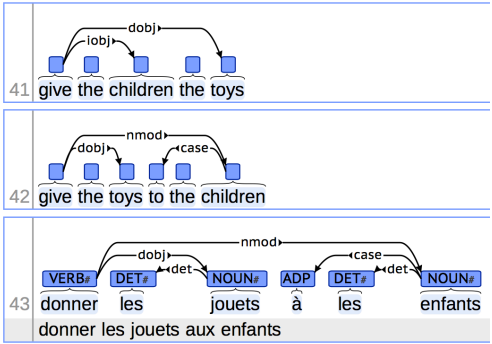


- A theory behind structure of phrase as scaffolding for the sentence meaning
- ...in linguistics
- ...in programming languages
- Main difference?
  - Ambiguity

Dependency graphs for sentences







- Simple semantic relations: Who did what to whom, where, why...
- Popular in NLP: Universal Dependencies
  - https://universaldependencies.org/
  - Originally annotated **by hand**
- This talk: Similar graphs built **automatically based on syntax theory**

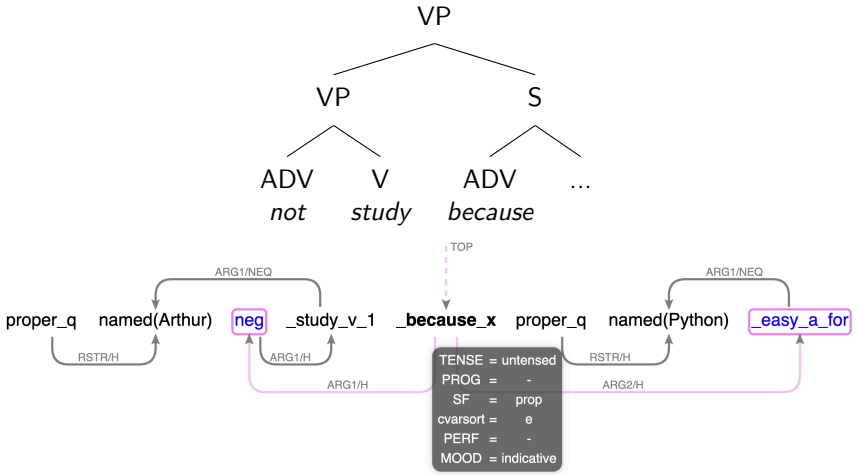
Ambiguous Syntactic and Semantic structure: Scope of negation

Arthur does not study because Python is easy

- Does Arthur study?
- Why?

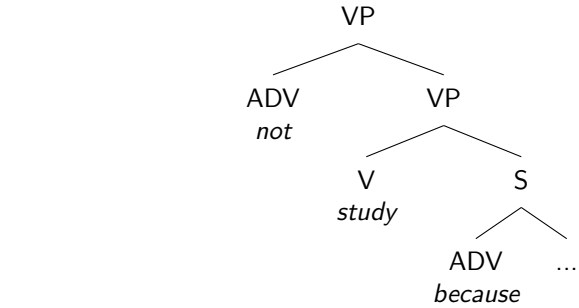
Ambiguous Syntactic and Semantic structure: Scope of negation

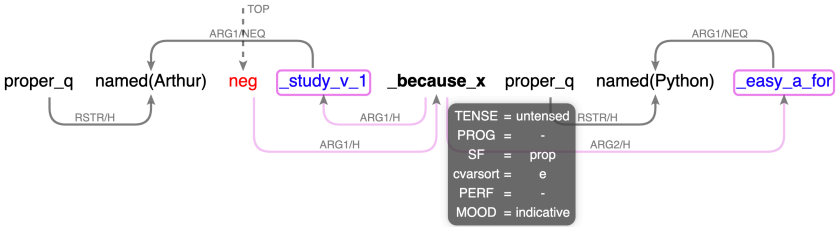
Arthur [does not study] [because Python is easy]



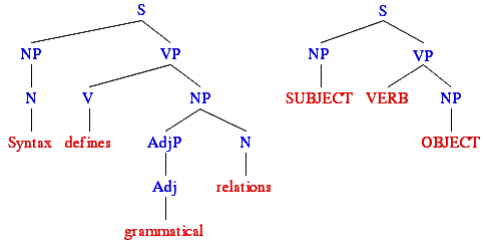
Ambiguous Syntactic and Semantic structure: Scope of negation

Arthur does not [study because Python is easy]



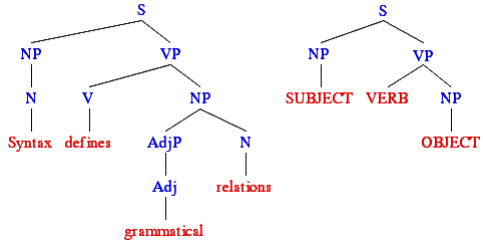


Why do we study syntax?



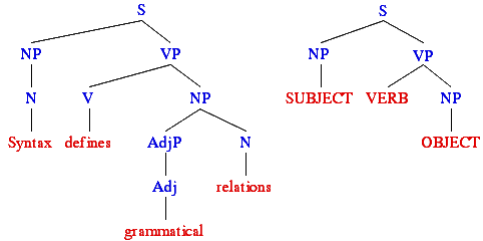
- To understand fundamental properties of languages

Why do we study syntax?



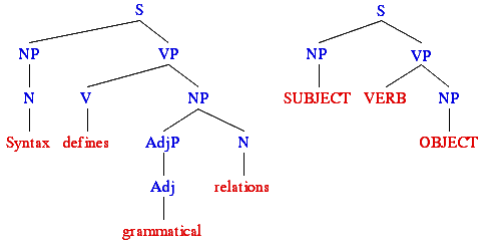
- To understand fundamental properties of languages
  - Out of curiosity

Why do we study syntax?



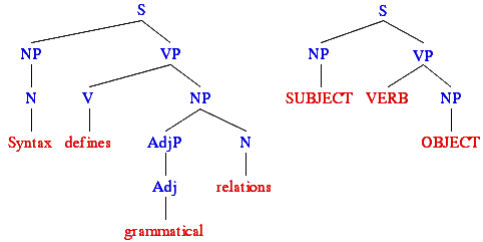
- To understand fundamental properties of languages
  - Out of curiosity
  - To be able to solve language-related problems in a more general way

Why do we study syntax?



- To understand fundamental properties of languages
  - Out of curiosity
  - To be able to solve language-related problems in a more general way
    - Imagine writing a compiler without knowing the full syntax

Why do we study syntax?



- To understand fundamental properties of languages
  - Out of curiosity
  - To be able to solve language-related problems in a more general way
    - Imagine writing a compiler without knowing the full syntax
    - But don't grammar teachers already know all about it?

Prescripive and descriptive grammar

- Prescriptive:
  - What is **prestigious** and what is not?
  - *I ain't got no cash*
- Descriptive:
  - What is **possible** and what is not?
  - *Cash no I got

Methodology: How to obtain a grammar?

Rafael's School of Athens (*circa* 1510)

- Formal (write by hand)
- Statistical (automatic learning)

Methodology: How to obtain a grammar?

Rafael's School of Athens (*circa* 1510)

- Formal (write by hand)
- Statistical (automatic learning)
- **Formal** is to **statistical** how **idealist** is to **materialist**

Methodology: How to obtain a grammar?

Rafael's School of Athens (*circa* 1510)

- Formal (write by hand)
- Statistical (automatic learning)
- **Formal** is to **statistical** how **idealist** is to **materialist**
- Ancient debate

Methodology: How to obtain a grammar?

Rafael's School of Athens (*circa* 1510)

- Formal (write by hand)
- Statistical (automatic learning)
- **Formal** is to **statistical** how **idealist** is to **materialist**
- Ancient debate
- Computational syntax today uses **formal** methods

PyDelphin library

- DELPH-IN is an international grammar engineering research consortium
  - Grammar engineering for syntactic theory
  - 2023 summit was at A Coruña
  - https://github.com/delph-in/docs/wiki
  - https://delph-in.github.io/delphin-viz/demo

PyDelphin library

- DELPH-IN is an international grammar engineering research consortium
  - Grammar engineering for syntactic theory
  - 2023 summit was at A Coruña
  - https://github.com/delph-in/docs/wiki
  - https://delph-in.github.io/delphin-viz/demo
  - Large grammars of English, Spanish, Chinese, Japanese
  - Smaller grammars for many other languages

PyDelphin library

- DELPH-IN is an international grammar engineering research consortium
  - Grammar engineering for syntactic theory
  - 2023 summit was at A Coruña
  - https://github.com/delph-in/docs/wiki
  - https://delph-in.github.io/delphin-viz/demo
  - Large grammars of English, Spanish, Chinese, Japanese
  - Smaller grammars for many other languages
  - Systems for precision parsing and generation

PyDelphin library

- DELPH-IN is an international grammar engineering research consortium
  - Grammar engineering for syntactic theory
  - 2023 summit was at A Coruña
  - https://github.com/delph-in/docs/wiki
  - https://delph-in.github.io/delphin-viz/demo
  - Large grammars of English, Spanish, Chinese, Japanese
  - Smaller grammars for many other languages
  - Systems for precision parsing and generation
  - Using C, Lisp, Java, C#, Perl... Over the last 20 years

PyDelphin library

- DELPH-IN is an international grammar engineering research consortium
  - Grammar engineering for syntactic theory
  - 2023 summit was at A Coruña
  - https://github.com/delph-in/docs/wiki
  - https://delph-in.github.io/delphin-viz/demo
  - Large grammars of English, Spanish, Chinese, Japanese
  - Smaller grammars for many other languages
  - Systems for precision parsing and generation
  - Using C, Lisp, Java, C#, Perl... Over the last 20 years
- PyDelphin is a Python library for working with DELPH-IN data
  - https://pydelphin.readthedocs.io/
  - https://github.com/delph-in/pydelphin/graphs/contributors
  - **Python has significantly sped up research in all areas including syntax**

PyDelphin

- Extensive documentation
- Regression tests
- public APIs
- command line interface
- PEP-420 implicit namespaces
  - https://pydelphin.readthedocs.io/en/latest/guides/developer.html
- in-memory database transactions
  - complex parsed structures are stored in relational databases
- regular expressions preprocessor (REPP)
- parser/generator wrappers

Demo: command line

- Minimal toolkit:
  - The English Resource Grammar
    - https://github.com/delph-in/erg
  - The ACE parser
    - https://github.com/delph-in/docs/wiki/AceInstall
- echo "I did no study because Python is easy." | ace -g /Research/ERG/trunk/ace/erg.dat -1 | delphin convert –from ace –to dmrs-penman

A note on Visualization

- not the main purpose
- related graph visualization package Penman
  - https://penman.readthedocs.io/
  - tikz, LaTeX
- https://delph-in.github.io/delphin-viz/demo/

Concluding Remarks

- Demo continues in PyCharm
- DELPH-IN: DEep Language Processing with HPSG INitiative
- HPSG: A theory of syntax proposed by Pollard and Sag (1994)

Demo: Parser response object

- from delphin import ace as prs
- grm = "/Users/olzama/Research/ERG/trunk/ace/erg.dat"
- response = prs.parse(grm, "I did not study because Python is easy.")
- NOTE: parsed 1 / 1 sentences, avg 14744k, time 0.81425s

Demo: structures contained in the response

- from delphin import dmrs
- m = response.result(0).mrs()
- d = dmrs.from_mrs(m)

Demo: databases

- from delphin import itsdb
- ts = itsdb.TestSuite('/Users/olzama/Research/ERG/trunk/tsdb/gold/mrs')
- len(ts['item'])
- ts['item'][0]['i-input']
- ts['item'].update(0, 'i-input': 'It snowed.')
- ts['item'][0]['i-input']

Demo: Parsing databases

- with prs.ACEParser(grm) as cpu:
  - ts.process(cpu)
- NOTE: parsed 107 / 107 sentences, avg 5410k, time 2.40371s

Demo: Accessing parse databases

- from delphin import tsql
- selection = tsql.select('i-id i-input where i-length > 5 readings > 0', ts)
- next(iter(selection))
- ('61', 'Abrams handed the cigarette to Browne.')

Linguistics, Computational linguistics, NLP...

- Linguistics: Study of systematicity in **human** language, social science
  - NB: not the same as philology

Linguistics, Computational linguistics, NLP...

- Linguistics: Study of systematicity in **human** language, social science
  - NB: not the same as philology
- Computational linguistics:

Linguistics, Computational linguistics, NLP...

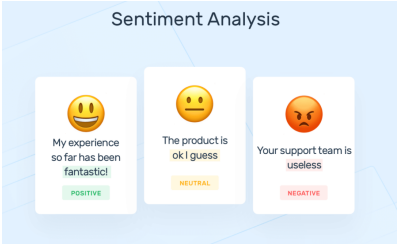
- Linguistics: Study of systematicity in **human** language 🌏, social science
  - NB: not the same as philology
- Computational linguistics:
  - Applying computational methods to linguistics
    - a subfield of **linguistics**

Linguistics, Computational linguistics, NLP...

- Linguistics: Study of systematicity in **human** language, social science
  - NB: not the same as philology
- Computational linguistics:
  - Applying computational methods to linguistics
    - a subfield of **linguistics**
  - Natural language processing (NLP)
    - a subfield of **artificial intelligence**
- Why the same term?

Natural language processing

- How can machine reasoning be compared to human reasoning?
- What can be learned about the world through the lens of language data? (Yatskar, p.c.)
- Task-oriented, due to $$ from industry, but now focusing more on RQ



https://www.analyticsvidhya.com/blog/2022/01/sentiment-analysis-with-lstm/

Linguistics (computational or not)

- What are some natural laws in how people acquire languages?
- What are **people's attitudes** to various language behaviors?
- In what ways are languages of the world **fundamentally** similar?
  - Syntactic Theory

https://languagelog.ldc.upenn.edu/nll/?p=29480