ETH zürich

Introduction to Natural Language

Ryan Cotterell

Structure of this Lecture

1. Introduction
2. A Bit on Linguistics
3. What is NLP?
4. Course Logistics

Supplementary Material

Reading: Eisenstein Ch. 1

Warning: This course is brand new

- First time this class being taught at ETHZ
  - Yay! 🎉
- ETHZ previously offered Natural Language Understanding, which had a very different syllabus and focus
- We are making the slides (largely) from scratch
  - Please give us feedback!

What to do if you spot a typo?

- Help us make the lectures better!
- Slides posted after the lecture (with typo fixes)
- Please email ryan.cotterell@inf.ethz.ch or post on Piazza

Two Important Websites for the Course

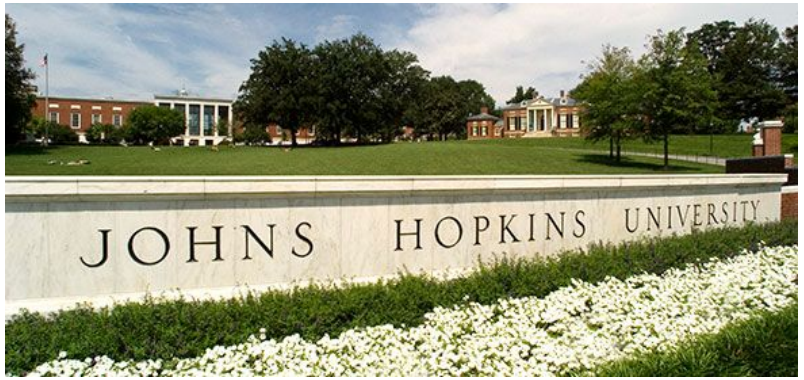
- Course website: https://rycolab.github.io/classes/intro-nlp/
  - This will be regularly updated with important information

- Piazza: https://piazza.com/class/kei3py4i26c4jw
  - You do not have to sign up for Piazza and all critical issues will be posted elsewhere
  - However, learning through group discussion will help you better understand the content of the course

Who am I?

- My name is Ryan (Cotterell)
  - No need to for titles, Ryan is fine

- I conduct research at ETH in a variety of subjects
  - computational linguistics
  - natural language processing
  - machine learning



- BA/MSc/PhD from John**s** Hopkins University
  - located in my beautiful hometown of Baltimore, MD
- Spent a year as a Lecturer at the University of Cambridge

Meet the Teaching Assistants

- Clara Meister (Head TA)
  - BSc/MSc from Stanford University
  - Despite the last name, my German ist sehr schlecht
- Niklas Stoehr
  - Germany → China → UK → **Switzerland** (first day today)
  - I like interdisciplinarity: NLP meets political and social science
- Pinjia He
  - PhD from The Chinese University of Hong Kong
  - Focus: robust NLP, NLP meets software engineering
- Francesco Varini
  - MSc at ETH (currently)
  - Studies focus: Machine Learning, Thesis in NLP

ETH zürich

Natural Language Processing is Everywhere!

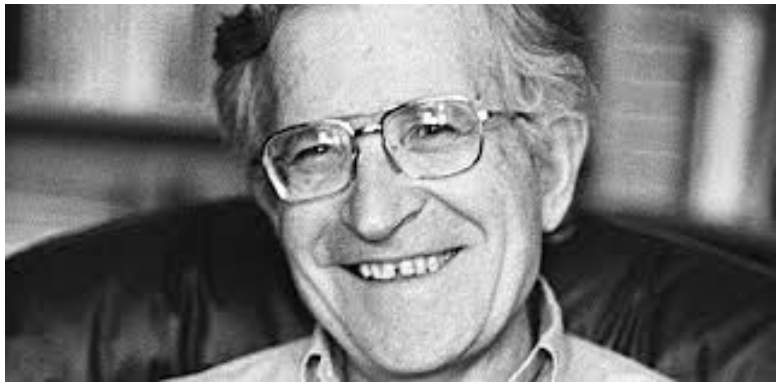
Natural Language Processing is Everywhere!

- The Association for Computational Linguistics (ACL) was founded in 1962
  - In the 1970s the conferences had < 100 participants
  - EMNLP 2018 had > 2500 participants
- NLP is the backbone of many major tech companies

The Statistical Revolution in NLP

In the early 90s, our field went from intuition-driven to data-driven

Noam Chomsky



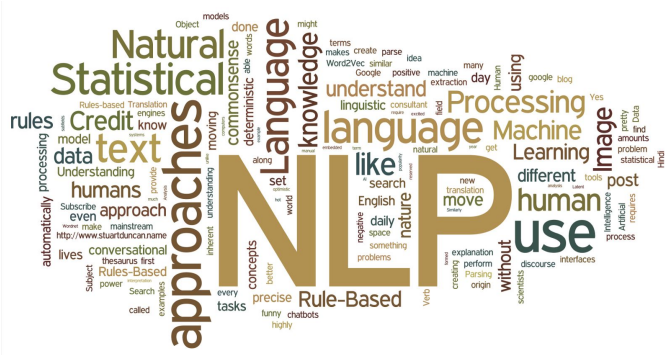
"But it must be recognized that the notion "probability of a sentence" is an entirely useless one, under any known interpretation of this term" (1969, p. 57)

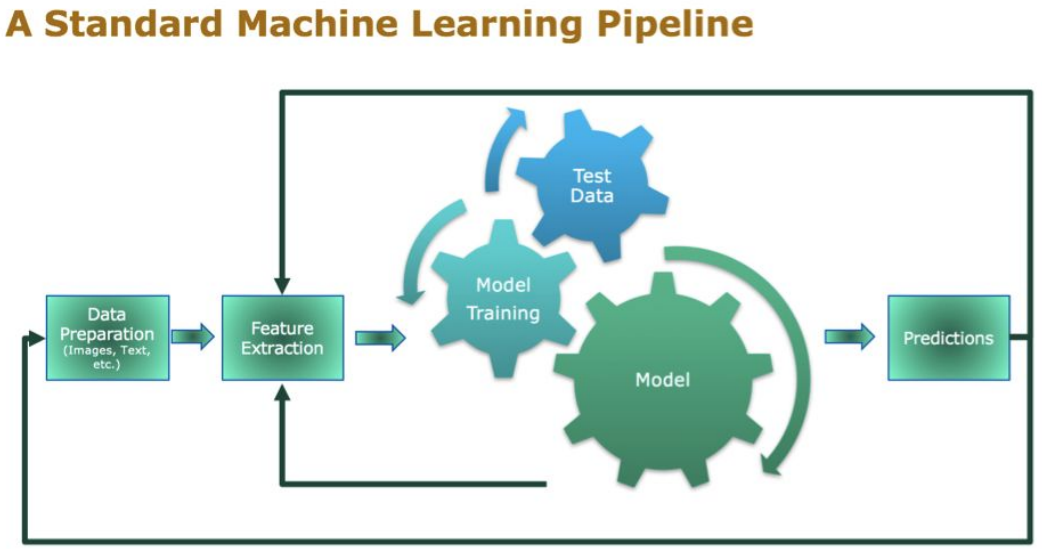
Fredrick Jelinek

"Anytime a linguist leaves the group the recognition rate goes up." (1988)

(Many) NLP Problems are treated as ML Problems

- NLP practitioners started annotating data and training statistical models
- Developed many consumer-facing products
- Machine translation, automatic speech recognition, document summarization

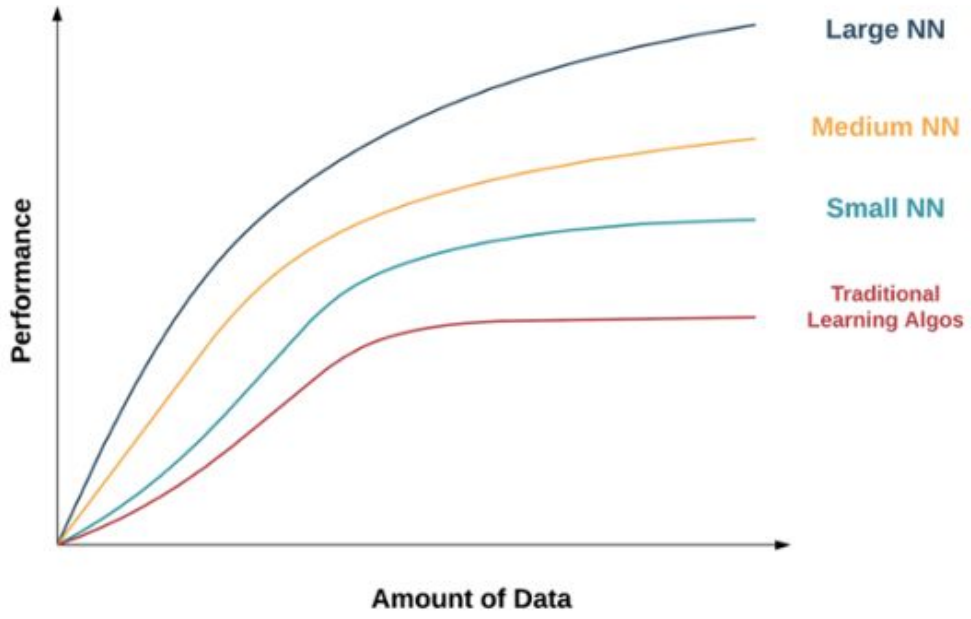




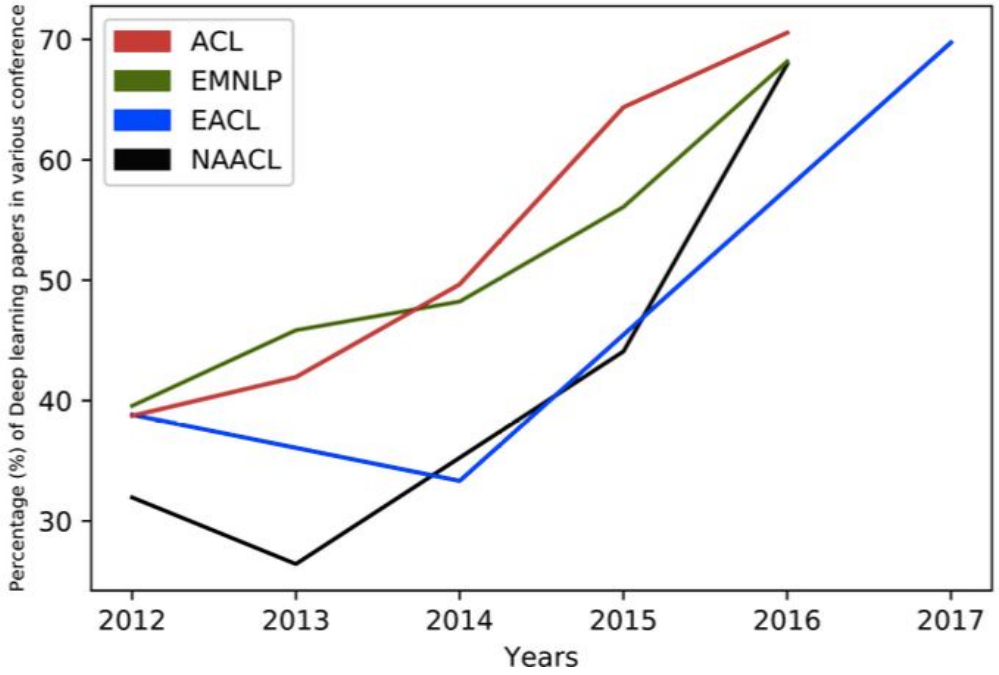
The Rise of Deep Learning in NLP

Use neural networks (with GPUs) to train bigger models on more data!

More Data, Better Performance



Dominates Top Venues



ETH zürich

A Bit on Linguistics (as a Science)

Human language is awesome!

- How human language is structured is one of the great unsolved scientific mysteries

- Despite the complexity of natural language, most students aren't exposed to formal descriptions until they reach university

- Students learn formal models of many natural phenomena
  - Falling objects (Newton's laws)
  - Electricity and magnetism (Maxwell's equations)
  - Evolution (Darwin's theory)

- Linguistics is defined as the ***scientific*** study of language
  - Human language is natural—ergo, linguistics is a ***natural*** science

Linguistics is a Weird Science

- You are all here are at one of the greatest technical universities in the world

- ETHZ offers courses in many natural sciences:
  - biology
  - chemistry
  - physics
  - economics

- But no linguistics! Why not?

What is linguistics?

- Traditionally, linguistics was classified as a humanity
  - There's an old joke that linguistics is the only humanity where the best training for graduate school is an undergraduate degree in math!

- Noam Chomsky took a lot of graduate-level abstract algebra that enabled his research
- Indeed, the Chomsky-Schützenberger theorem is a famous result in the study of formal power series

- **Digression**: How do you pronounce Schützenberger?

tion from a point of view intermediate between the two just mentioned. We will consider a representation of a language not as a set of strings and not as a set of structural descriptions, but as a set of pairs (σ, n), where σ is a string and n expresses its degree of ambiguity; that is, n is the number of different structural descriptions assigned to σ by the grammar G generating the language to which it belongs.

2. Grammars as Generators of Formal Power Series

2.1. Suppose that we are given a finite vocabulary V partitioned into the sets V_T (= terminal vocabulary) and V_N (= non-terminal vocabulary). We consider now languages with the vocabulary V_T, and grammars that take their non-terminals from V_N. Let $F(V_T)$ be the free monoid generated by V_T, i.e., the set of all strings in the vocabulary V_T. A language is, then, a subset of $F(V_T)$.

Consider a mapping r which assigns to each string $f \in F(V_T)$ a certain integer $\langle r, f\rangle$. Such a mapping can be represented by a *formal power series* (denoted also by r) in the non-commutative variables x of V_T. Thus

$$(8) \qquad r = \sum_i \langle r, f_i\rangle f_i = \langle r, f_1\rangle f_1 + \langle r, f_2\rangle f_2 + \ldots,$$

where $f_1, f_2, \ldots$ is an enumeration of all strings in V_T. We define the support of r (= Sup(r)) as the set of strings with non-zero coefficients in r. Thus

$$(9) \qquad \mathrm{Sup}(r) = \{f_i \in F(V_T) \mid \langle r, f_i\rangle \neq 0\}.$$

We do not insist that the coefficients $\langle r, f_i\rangle$ of the formal power series r in (8) be positive. If, in fact, for each i, $\langle r, f_i\rangle \geq 0$, then we shall say that r is a *positive* formal power series.

If for each $f_i \in F(V_T)$, the coefficient $\langle r, f_i\rangle$ is either zero or one, we say that r is the *characteristic* formal power series of its support.

2.2. If r is a formal power series and n an integer, we define the product nr as the formal power series with coefficients $\langle nr, f\rangle = n\langle r, f\rangle$, where $\langle r, f\rangle$ is the coefficient of f in r. Where r and r' are formal power series, we define $r + r'$ as the formal power series with coefficients $\langle r + r', f\rangle = \langle r, f\rangle + \langle r', f\rangle$, where $\langle r, f\rangle$ and $\langle r', f\rangle$ are, respectively, the coefficients of f in r and r'. We define rr' as the formal

Linguistics is a Weird Science

- Many branches of linguistics are in fact just mathematics
  - Much work in formal language theory in the US takes place in linguistics departments
  - In contrast, research in formal language theory is still common in CS departments in Europe

- However, most (?) incoming in linguistic PhD students have never taken a college-level math course!

- But, linguistics is a largely (non-statistical) mathematical modeling discipline
  - This leads to a weird scientific field of mathematical modeling

How does one do the science of language?

- In society, we are brought up with the idea that biology, chemistry and physics are sciences
- We are also brought up with the idea that literature, history and art are not
- Many of you (most?) will never have thought of mathematically modeling language
  - Where do we start?
- To give a taste of the subject, we zoom in on modeling one aspect of language: grammaticality

The Grammaticality Problem

- Let's analyze the **grammaticality problem**
  - The fundamental problem in syntax is the study of grammaticality

- **Basic Fact**: Some sentences are grammatical and some are not
  - Caveat: Depends on the dialect and individual speaker

- Humans tend to have strong judgements and they tend to be binary (Chomsky 1957)
  - A lot more to be said here, e.g. acceptability judgements, but beyond the scope of this class (Schütze 2016; Sprouse and Almeida 2017)

- When you learn grammar in school, you focus on **prescriptivism** and avoiding common "grammatical mistakes"
  - Prescriptivism is the practice of forcing people to obey (at times arbitrary) rules
  - E.g, don't a sentence in a preposition (common prescription in English)

A Brief Detour: Competence versus Performance

- A important dichotomy is the distinction between **competence** and **performance**

- Grammatical competence is a platonic ideal
  - There exists a true grammar of the language
  - Humans have imperfect knowledge or ability to execute this grammar

- Grammatical performance is a realizable thing
  - How do humans process and produce language
  - Psycholinguistics is the study of grammatical performance

- The competence–performance distinction is a very Chomksyan notion
  - Non-Chomksyans get upset about it and wont to do so
- The next few slides are about grammatical *competence* (not performance)

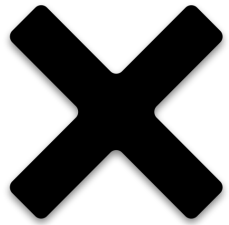
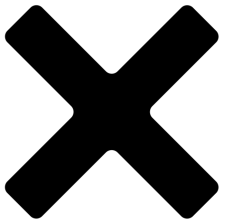
Some properties of the set of grammatical sentences

- The set is infinite, even if we have a finite lexicon
  - This is why language is often called the “infinite gift”
  - Due to recursion

- For now, let’s assume that we have a finite set of words: $\sum$ = {*Borislav*, *Frederica*, *programs*, *in*, *the*, *lab, with*}

- The grammaticality question: Determine the subset A $\subseteq \sum^*$ that a native speaker would consider grammatical

Which sentences are grammatical?

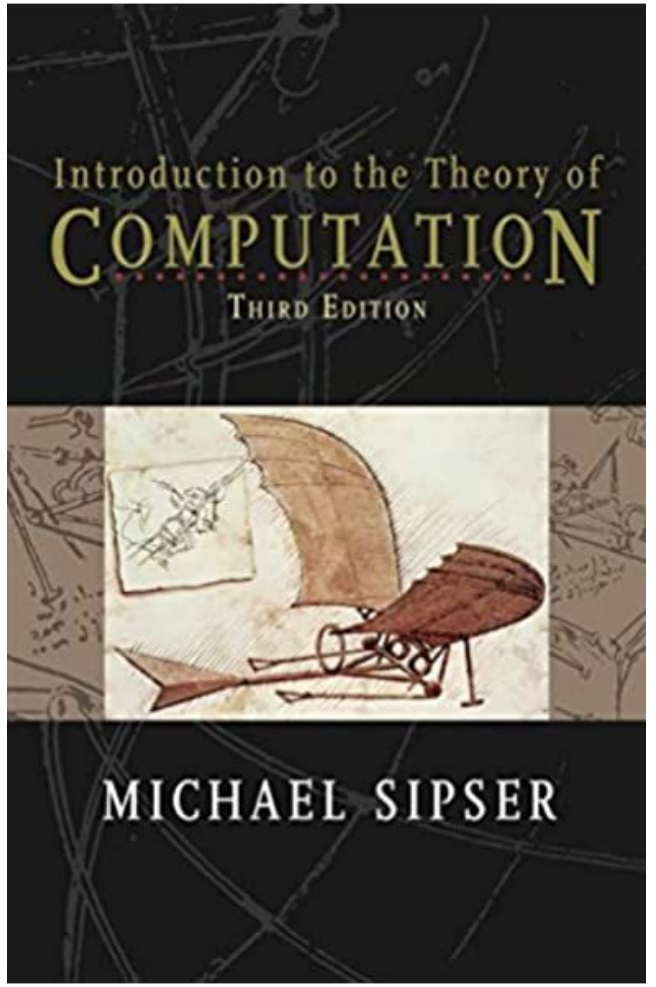
1. *Borislav lab programs* 
2. *Borislav programs in the lab* 
3. *Frederica programs in the lab with Borislav* 
4. *Borislav Borislav Borislav* 

A Formal Goal of Syntax

- Syntacticians develop the function that answer this question!

Mathematical Models of Sets of Strings

- This sounds like a familiar problem...
- Certainly, you have all seen modeling sets of $\sum^*$ before?
- That's right, in your theory of computation courses:
  - finite-state automata
  - context-free grammars
  - Turing machines
- Much of the machinery was developed for natural language
  - Chomsky's name is all over the field
  - Chomsky normal form, Chomsky hierarchy



Is Natural Language Finite-State?

- The short answer is “no.” But, why not?

- Are all grammatical subsets of {*Borislav*, *Frederica*, *programs*, *in*, *the*, *lab, with*} in English a finite-state set?
  - Yes!

- To get out of the finite-state world, we need center embedding
  - Other tricks will work as well, but this one is simplest

Is Natural Language Finite-State?

- Consider the following sentences (Jurafsky and Martin 2008)
  - [The cat [likes tuna fish]]
  - [The cat the dog [chased [likes tuna fish]]]
  - [The cat the dog the rat [bit [chased [likes tuna fish]]]]

- The brackets show (certain) constituent boundaries
  - Reminiscent of a Dyck language? Famously context free

- Noam Chomsky (1965) famously made a similar argument that language was not regular

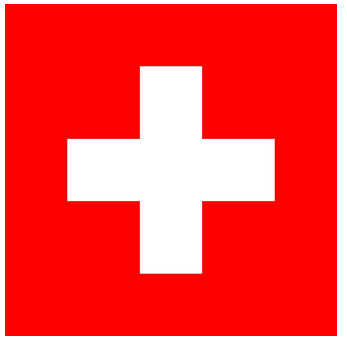
A Formal Proof of Non-Regularly of English

- Consider the following sentences (Jurafsky and Martin 2008)
  - [The cat [likes tuna fish]]
  - [The cat the dog [chased [likes tuna fish]]]
  - [The cat the dog the rat [bit [chased [likes tuna fish]]]]

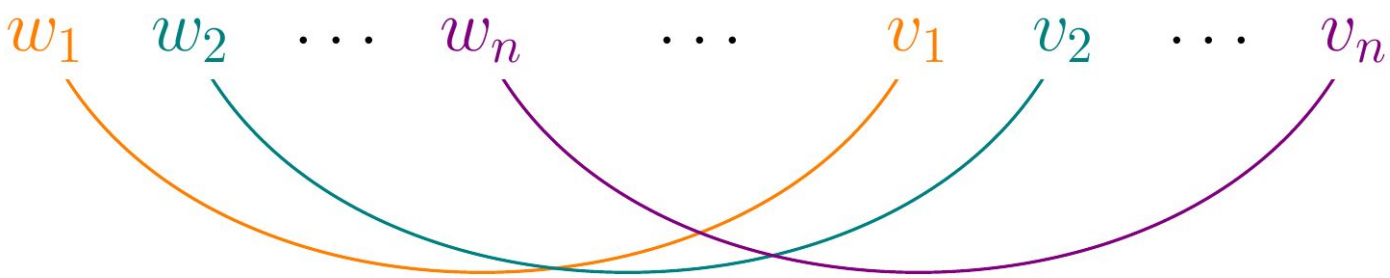
- The sentences above have the form:
  - $(\text{Noun Phrase})^{n}$ $(\text{Transitive Verb Phrase})^{(n-1)}$ likes tuna fish
  - For simplicity, let's write this as: $a^{n}\ b^{n-1}\ c$

- Easily shown to be non-regular by the pumping lemma
  - We would have to pump some combination of a's and b's

Is Natural Language Context-free?



- The famous counter-example is Swiss German (Schieber 1985)
- Cross-serial dependencies are not non-context free (wiki)



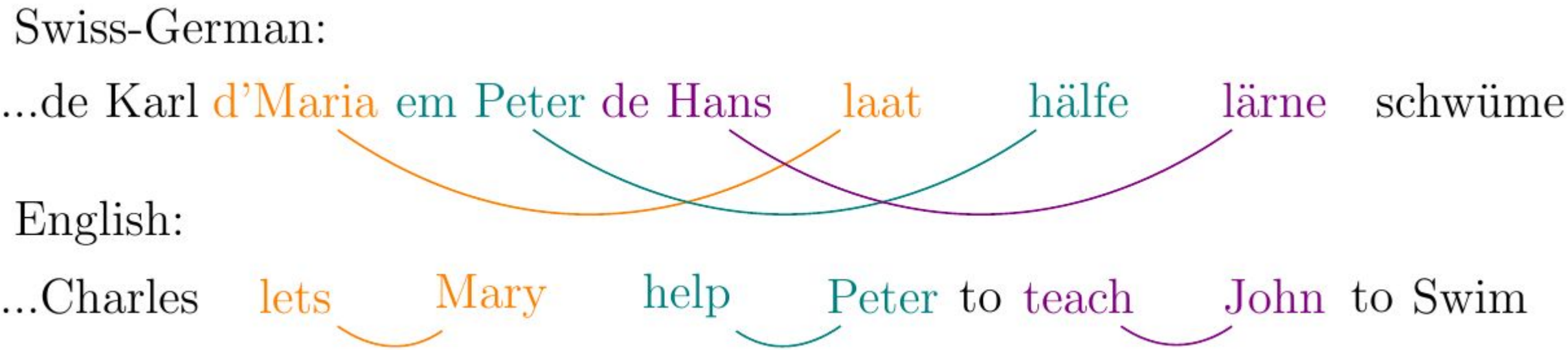
Swiss-German:

...mer em Hans es huss hälfed aastriiche

English:

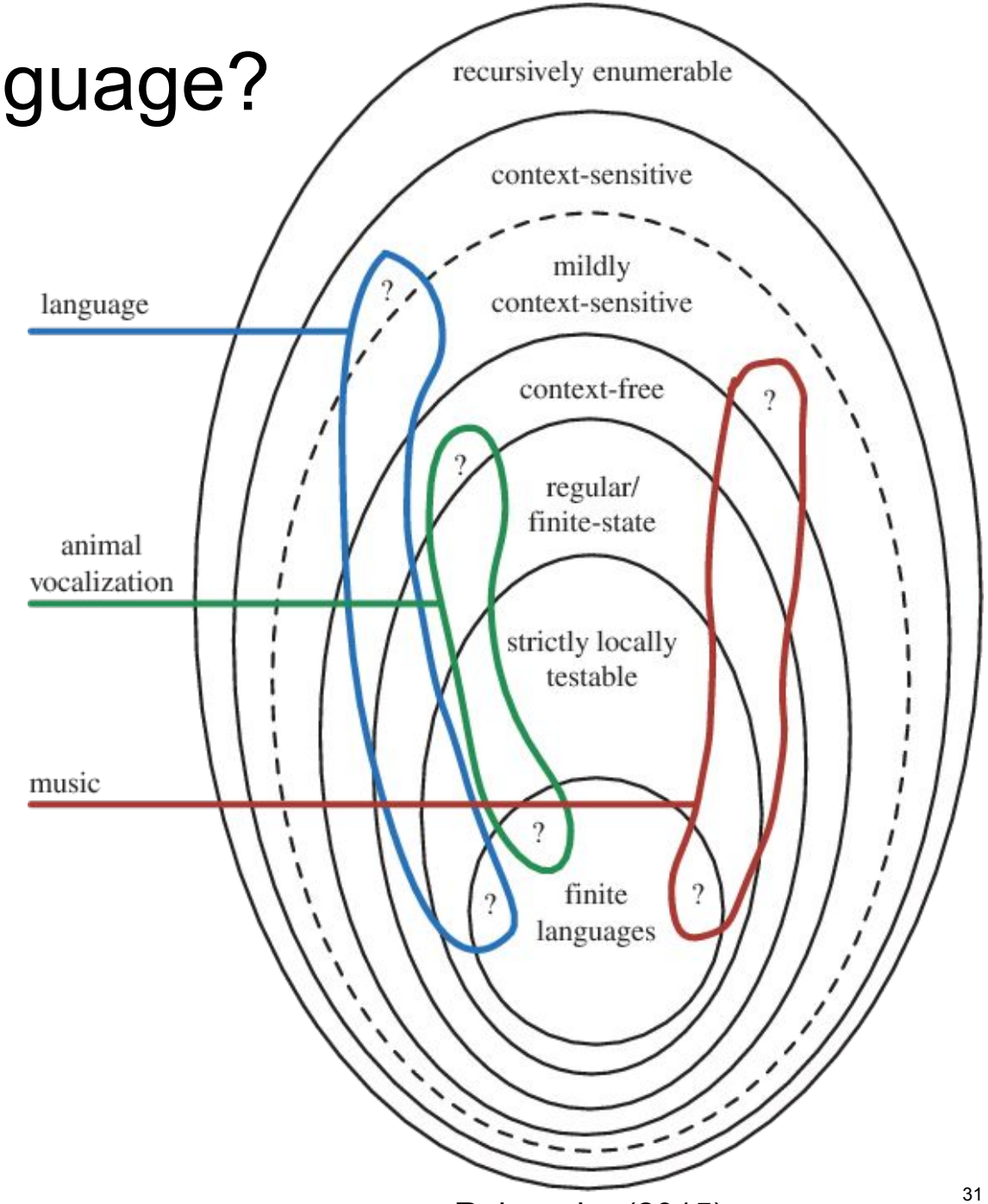
...we helped Hans paint the house

A More Complicated Example in Swiss German



How complex is human language?

- Many suspect natural language language syntax is mildly context sensitive
  - Still admit a polynomial-time recognize algorithm
  - Context-sensitive languages are generally only recognizable in exponential time
- Morphology (word building) is widely speculated to be regular
- Compare with animal sounds and music

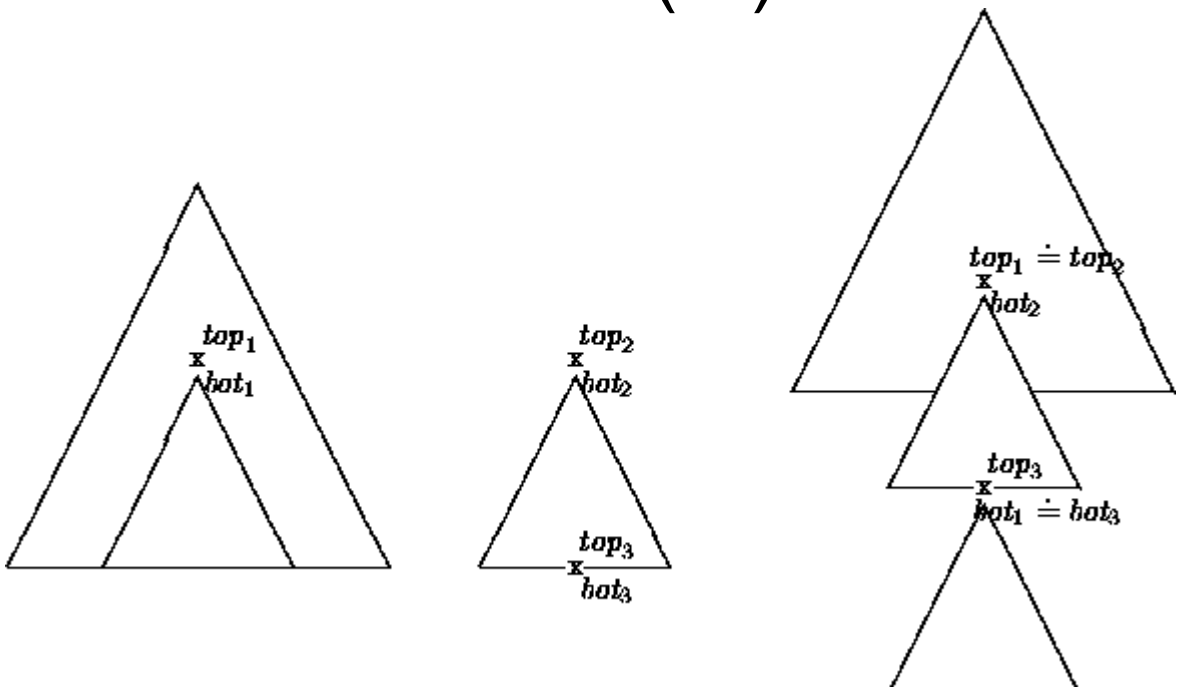


Rohrmeier (2015)

Mildly Context-Sensitive Formalisms

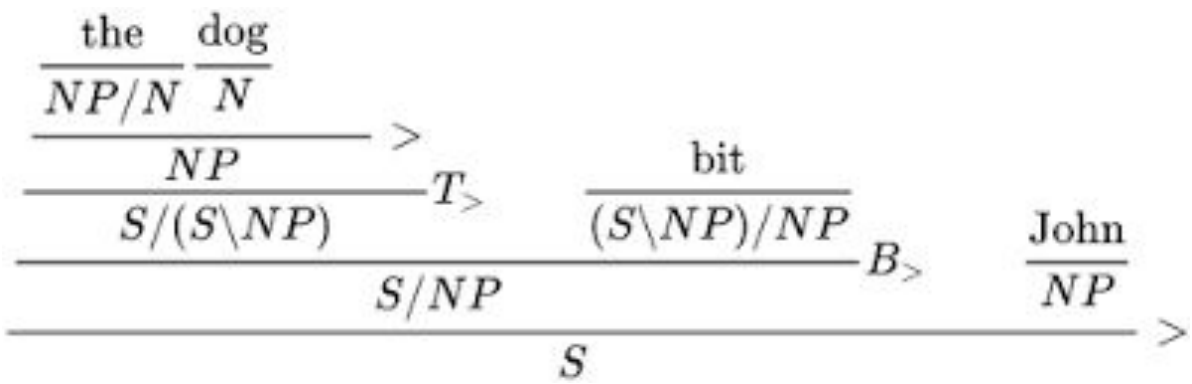
Tree-Adjoining Grammar

- CFG + adjunction
- Parsable in $O(n^6)$



Combinatory categorial grammar

- Generalization of slashed-category CFGs
- Parsable in $O(n^6)$



What do syntacticians do professionally?

- They formally study the grammaticality problem
  - Perhaps, a bit simplified, but largely accurate (in Ryan's opinion)

- They come up with sophisticated formal language theory
  - Demonstrate the adequacy of the formalisms for modeling known syntactic phenomena
  - Prove properties about the formalisms

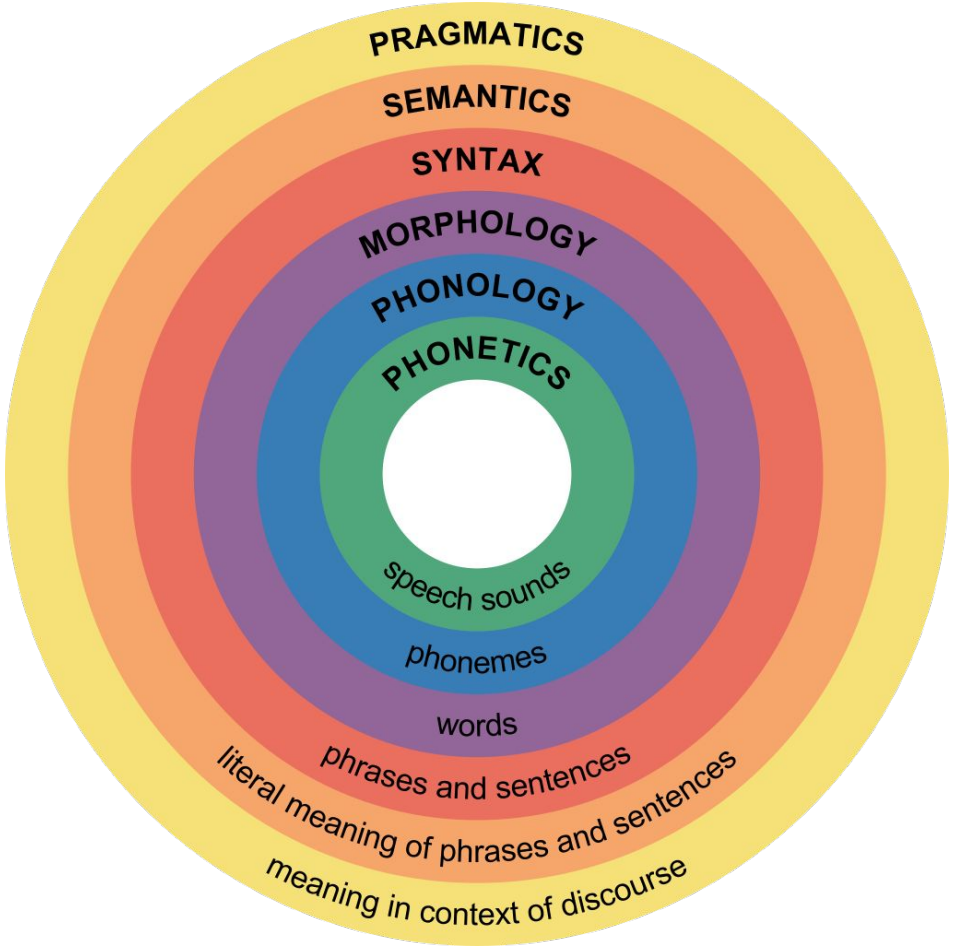
- Syntacticians also care about complexity:
  - Turing machine is too powerful!
  - What is the least complex formalism we could have?
  - Complexity ≈ level in Chomsky hierarchy
  - Occam's razor says choose the least complex formalism possible

Linguistics is more than Syntax

- The purpose of the proceeding slides was to ward off misconceptions about what linguistics is as science

- Linguists study all aspects of language
  - **phonology**: study of sound abstraction
  - **morphology**: study of word building
  - **syntax**: study of word order
  - **semantics**: study of meaning

- All of these sub-disciplines have similar mathematical techniques as syntax
  - Generally, not statistical



And Now for Something Completely Different...

- In contrast to linguistics, modern natural language processing has largely gone statistical
  - Most state-of-the-art models are large neural networks

- NLP is more than an applied ML discipline
  - There could come a day when the state-of-the-art approaches for NLP are no longer statistical

- NLP also has a different goal than linguistics
  - NLP: engineer systems to solve a problem
  - Linguistics: theorize about the structure of human language

ETH zürich

What is Natural Language Processing?

What is NLP?

- **Big picture:** A set of methods and algorithms for making natural languages accessible to computers

- **Motivation**:
  - **Analysis** (NL $\rightarrow \mathcal{R}$), e.g., topic classification
  - **Generation** ($\mathcal{R} \rightarrow$ NL), e.g., chat bots
  - **Acquisition** of $\mathcal{R}$ from knowledge and data, e.g., modeling

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

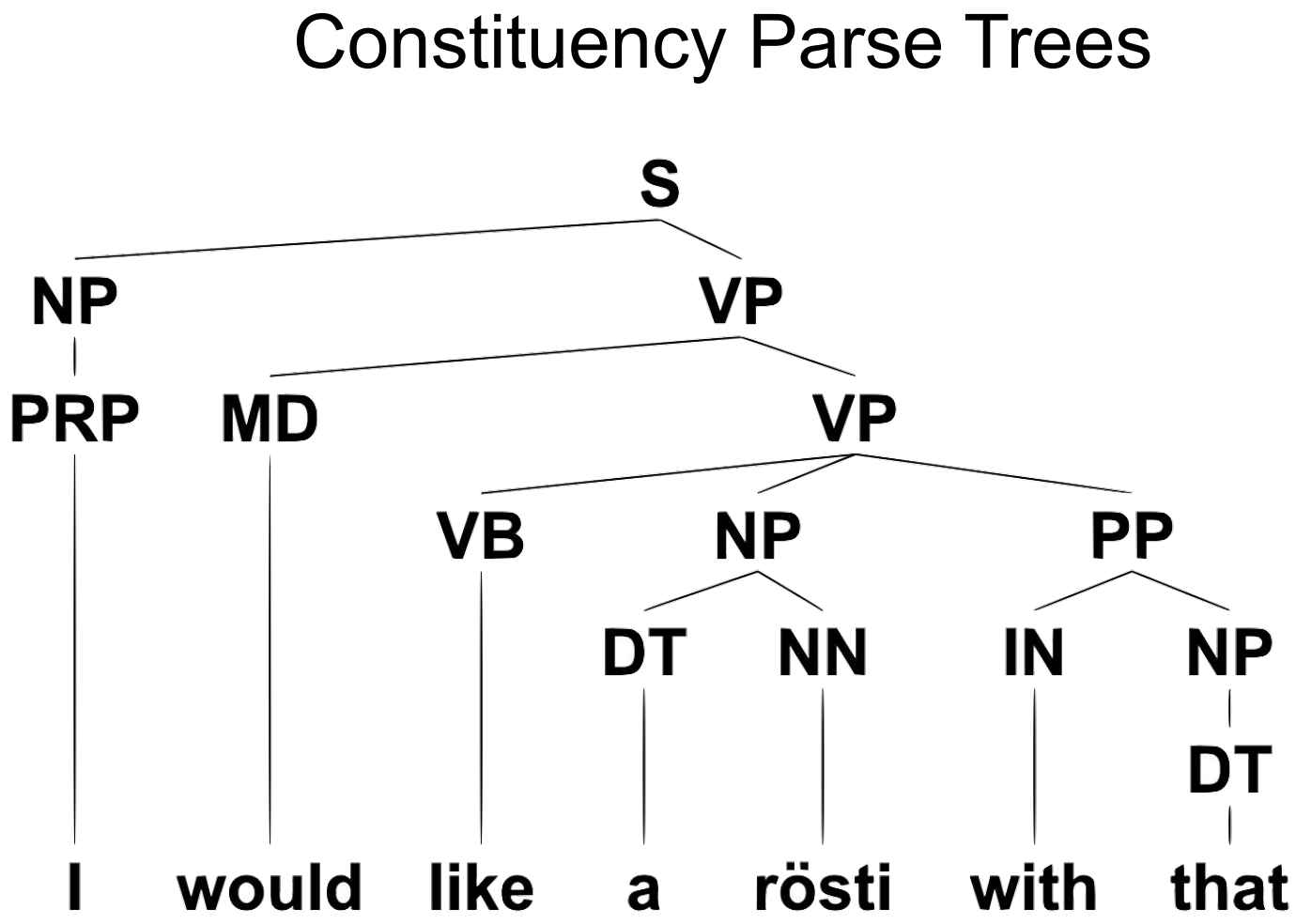
What is $\mathcal{R}$?

Big picture: some representation interpretable by a machine

- In some applications, we can liken $\mathcal{R}$ to a linguistic structure
  - Constituency parses (covered in lecture 7)
  - Dependency parses (covered in lecture 8)
- In some applications, we can liken $\mathcal{R}$ to “meaning”
  - Giving commands to a robot
  - Querying a database, e.g. construct a SQL query
  - Reasoning about relatively closed, grounded worlds
- In other cases, it is harder to describe
  - Analyzing opinions or sentiment (covered in lecture 4)
  - Identifying various biases in a text

Inspiration taken from Noah Smith’s CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Examples of $\mathcal{R}$



Examples of $\mathcal{R}$

Abstractive Meaning Representations

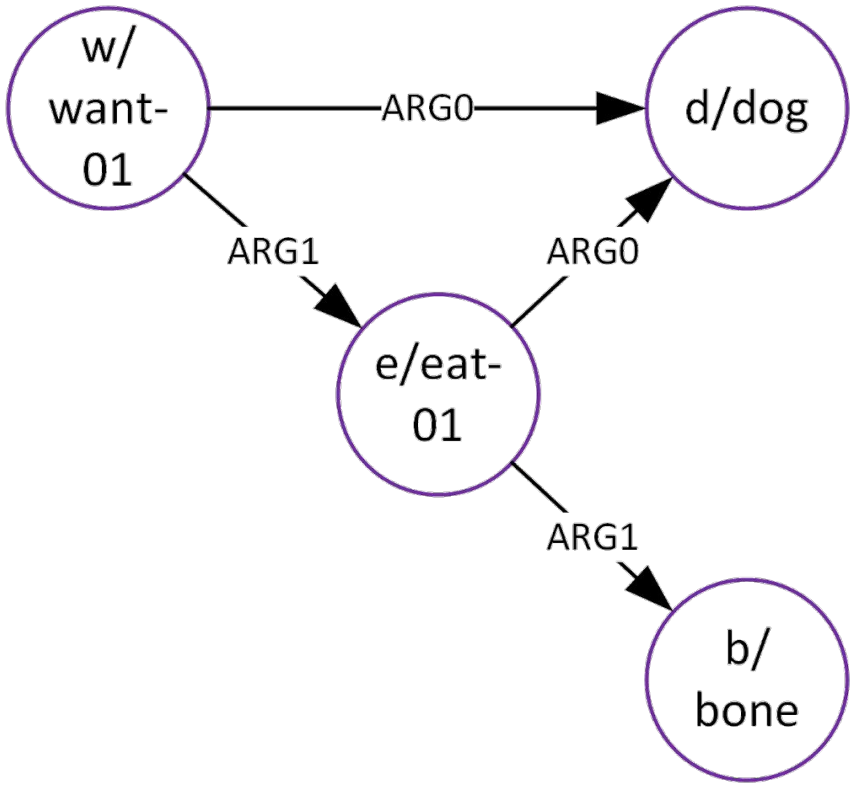
PENMAN Notation

$\exists$w, e, d, b
instance(w, want-01) $\wedge$ instance(d,dog)
$\wedge$ instance(e,eat) $\wedge$ intance(b,bone)
$\wedge$ arg0(w,d) $\wedge$ arg1(w,e)
$\wedge$ arg0(e,d) $\wedge$ arg1(e,b)

```
(w / want-01
    :ARG0 (d / dog)
    :ARG1 (e / eat-01
        :ARG0 d
        :ARG1 (b / bone)))
```

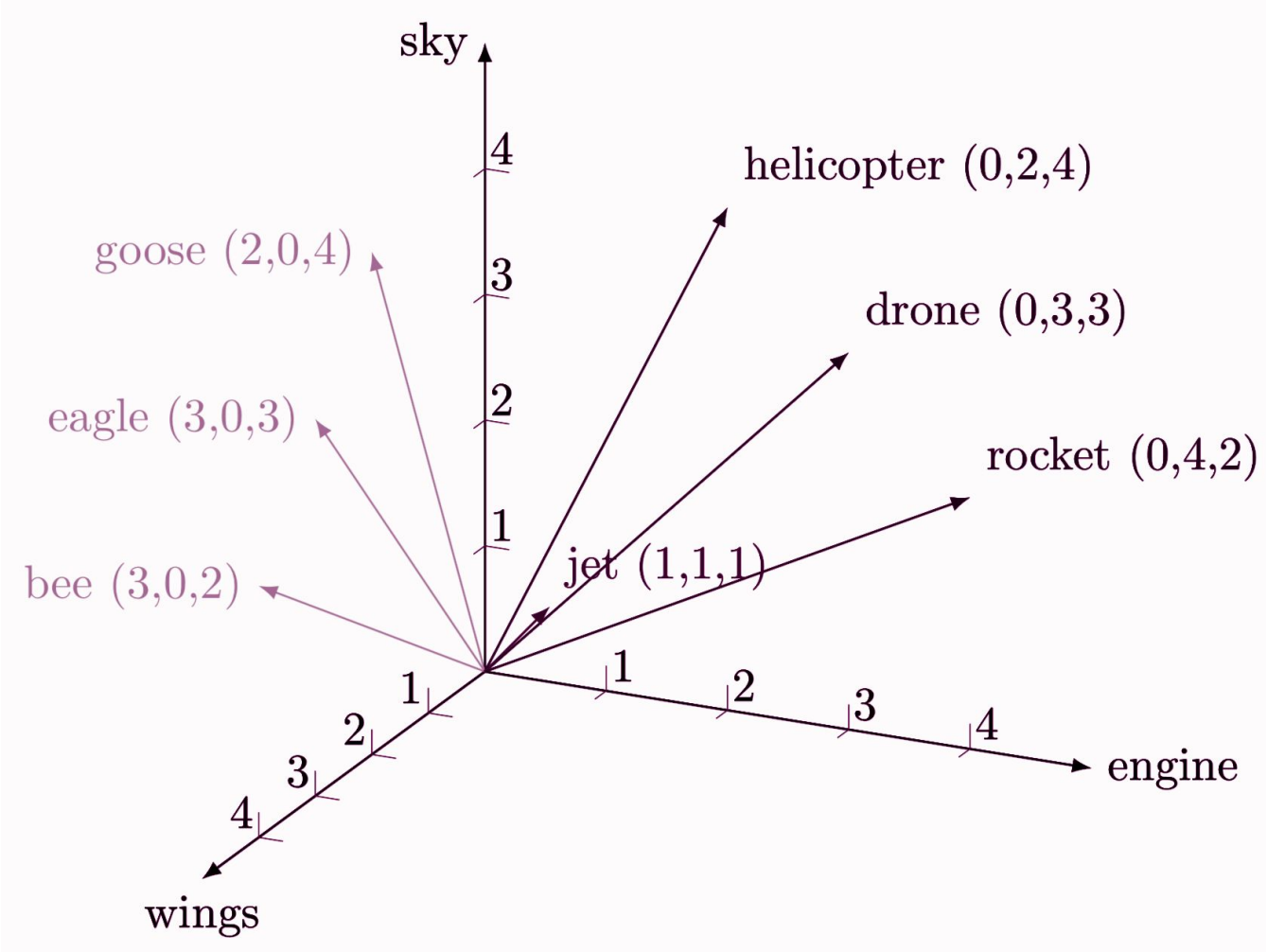
Conjunction Form

Graph Notation



Examples of $\mathcal{R}$

Word Embeddings



Why is NLP hard?

A few reasons:

1. Natural language is complex.
    - Meaning may be expressed many ways, and there are immeasurably many meanings.

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Semantics is Hard

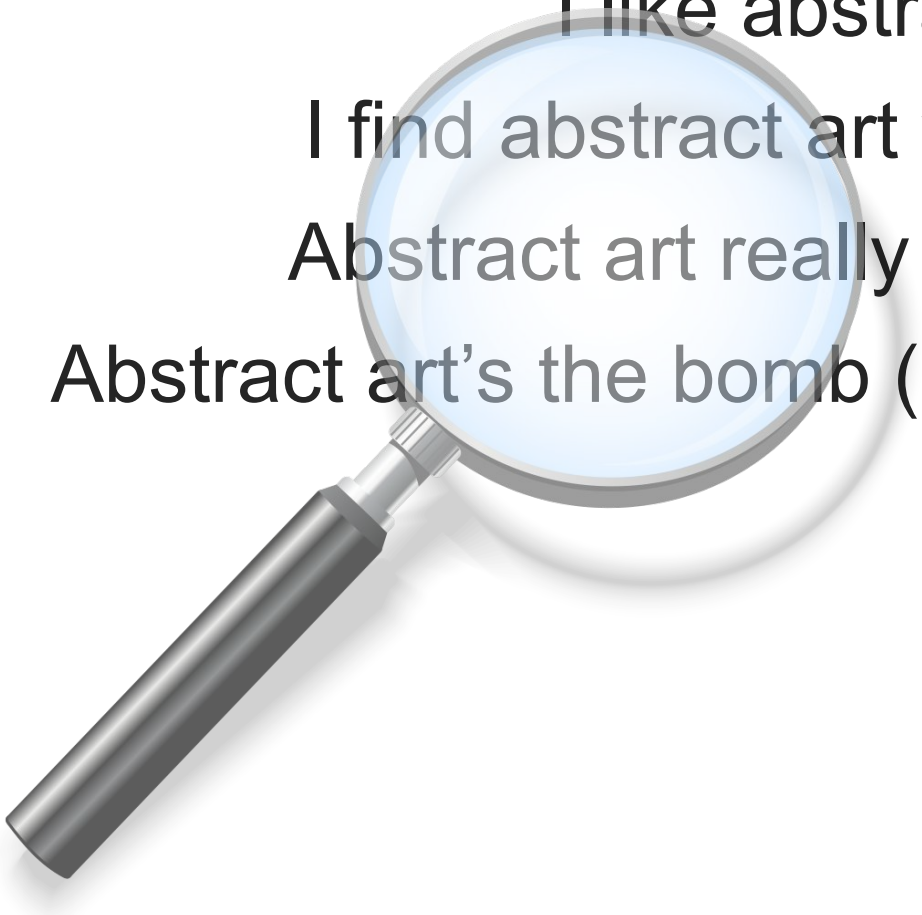
They all mean (more or less) the same thing!

I like abstract art

I find abstract art very pleasing

Abstract art really speaks to me

Abstract art's the bomb (90s American slang)



Why is NLP hard?

A few reasons:

1. Natural language is complex.
    - Meaning may be expressed many ways, and there are immeasurably many meanings.
    - A string can have many possible interpretations in different contexts; resolving ambiguity correctly may require external knowledge about the world.

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Examples

Syntax vs. Semantics

We saw the woman with the telescope wrapped in paper.

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Examples

Syntax vs. Semantics

We saw the woman with the telescope wrapped in paper.

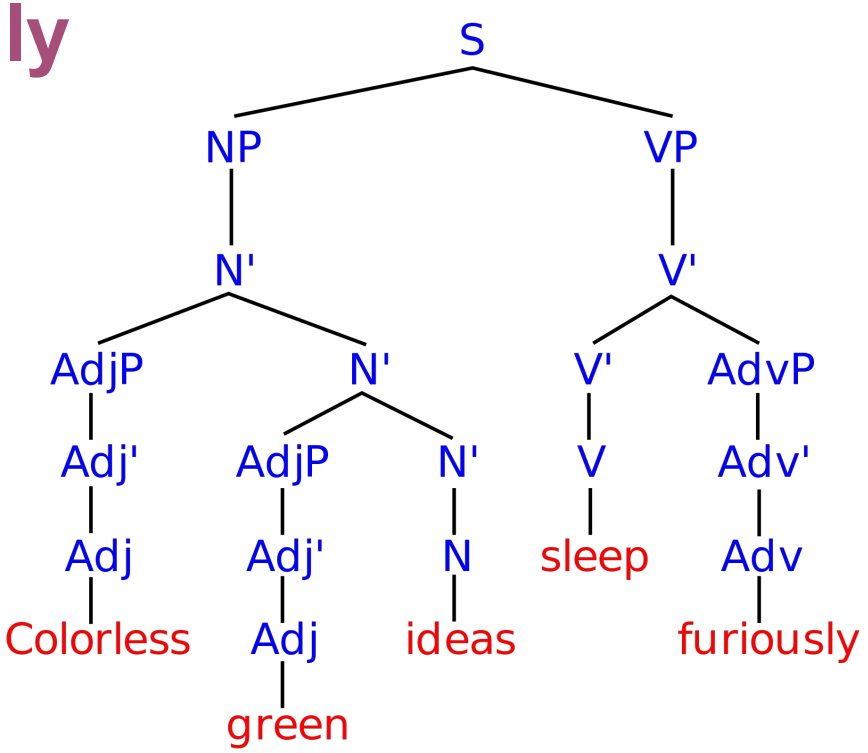
- Who has the telescope?
- Who or what is wrapped in paper?
- An event of perception, or a questionable attempt at assault?

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Examples

Syntax vs. Semantics

Colorless green ideas sleep furiously



Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Why is NLP hard?

A few reasons:

1. Natural language is complex.
    - Meaning may be expressed many ways, and there are immeasurably many meanings.
    - A string can have many possible interpretations in different contexts; resolving ambiguity correctly may require external knowledge about the world.
    - Linguistic diversity across languages, dialects, genres, styles, . . .

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Examples

Linguistic diversity

El café negro me gusta mucho

↓

The coffee black me pleases much

Examples

Linguistic diversity

El café negro me gusta mucho

↓

~~The coffee black me pleases much~~

I really like dark coffee

Why is NLP hard?

A few reasons:

1. Natural language is complex.
    - Meaning may be expressed many ways, and there are immeasurably many meanings.
    - A string can have many possible interpretations in different contexts; resolving ambiguity correctly may require external knowledge about the world.
    - Linguistic diversity across languages, dialects, genres, styles, . . .
2. Appropriateness of a representation depends on the application.
3. Any $\mathcal{R}$ is a theorized construct that involves bias in the associated method.
4. There are many sources of variation and noise in linguistic input.

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Applications of NLP Today

- Conversational agents
- Information extraction and question answering
- Machine translation
- Opinion and sentiment analysis
- Social media analysis
- Rich visual understanding
- Essay evaluation
- Mining legal, medical, or scholarly literature

How do we tackle NLP?

Engineering decisions under consideration:

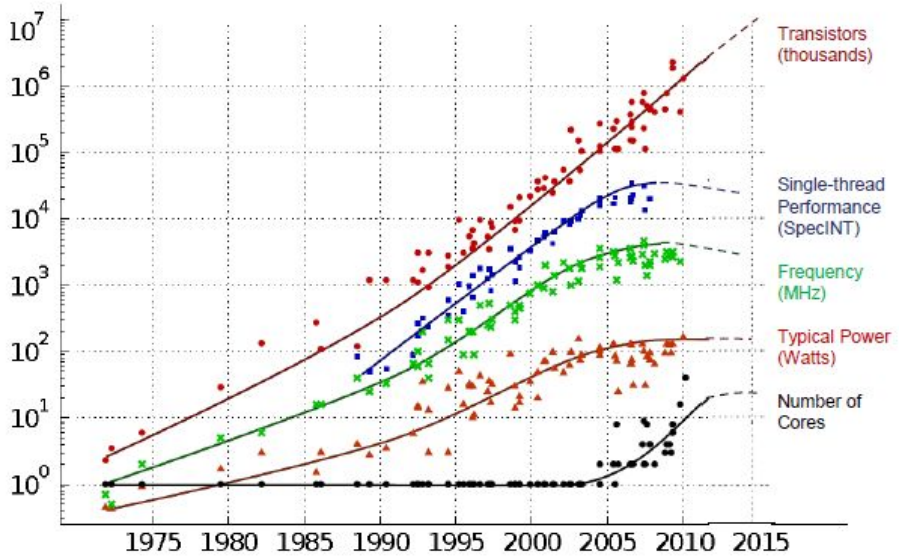
- Tasks are difficult to define formally; they are always evolving
- Objective evaluations of performance are always up for debate
- Different applications require different $\mathcal{R}$

Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

How do we tackle NLP?

External factors under consideration:

- Increases in computing power (Moore's Law)
- The spread of knowledge/data over the web and social media
- Advances in machine learning
- Advances in other related fields: linguistics, psycholinguistics, cognitive science...



Inspiration taken from Noah Smith's CSE 447 intro slides:: https://courses.cs.washington.edu/courses/cse447/19wi/slides/intro.pdf

Related Fields: Computational Linguistics

- Emphasis on “linguistics”; recall the beginning of lecture

- Field where computational methods aid in the goal of studying languages
- On the other hand, NLP describes methods for working with natural language as means to achieve another goal
  - Many (successful) NLP technique rely on the bias/assumptions from linguistic theory.

NLP is not just Machine Learning

- Overlap between the two fields: many contemporary NLP techniques use machine learning, however, NLP is ***not*** a subfield of machine learning
- NLP can employ a subset of machine learning methods:
    - Strings, unlike image or audio data, are discrete
    - Largely dealing with sequence and hierarchical data
- There are very successful non-statistical techniques
    - finite-state transducers for spell checking
    - rule-based syntactic parsers

ETH zürich

Course Logistics

Overview of the Course: Tentative Schedule

Lecture 1: Introduction to NLP

Lectures 2 and 3: Technical Components of NLP

- Backpropagation and log-linear modeling

Lectures 4 to 10: NLP tasks and methods for modeling them

- E.g., Sentiment analysis with multi-layer perceptrons, machine translation with Transformers

Lecture 11: Axes of Modeling

Lecture 12: Bias and Fairness in NLP

Schedule on the website: https://rycolab.github.io/classes/intro-nlp/

Two-Faceted Lectures

- Almost all the lectures in class are going to have two parts
  - The first part is going to be about a mathematical method
  - The second part is going to about a common task is NLP where that method is applicable

- Why this structure?
  - NLP classes at peer institutions primarily function as a literature review
  - But, the literature is always changing…
  - This focuses on fundamentals

- A neural architecture last for a conference season, but an algorithm is forever

Logistics about this Course

Meeting times

- Lecture: Monday 12 - 14 HG D1.2
- Discussion Section(?): Wednesday 13 - 14 HG D1.2

Materials

- Introduction to Natural Language Processing (Eisenstein) (https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf)

Grading

- One project (30% of the grade)
- Final exam (70% of the grade)

Piazza: https://piazza.com/class/kei3py4i26c4jw

Website: https://rycolab.github.io/classes/intro-nlp/

Course Project (30% of the grade)

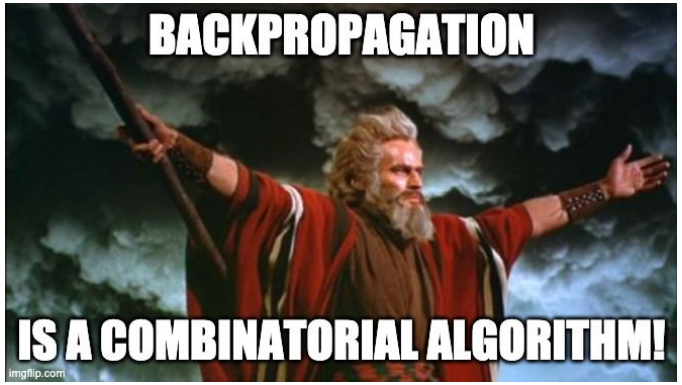
- A focused research projects conducted in groups of two to five
- The deliverables of the project will be
  - A working system for a task
  - A write-up that contains
    - A literature review
    - The mathematical details
    - Good experimental comparison to previous methods
- Like an ACL-style research paper but without a novelty requirement
  - We will even make you use the sty file :P

ETH zürich

Sneak Preview of Next Lecture

Backpropagation: What is it really?

- Many of the recent advances in neural network architectures would not be computationally feasible without backpropagation.
- Most people do not understand the full beauty of backpropagation, believing it to just be a simple instantiation of the chain rule
- However, backpropagation is much more than just the chain rule: it is a **dynamic program** that exploits the composite nature of complex functions to compute derivatives in **linear time**, where direct application of the chain rule would frequently lead to a combinatorial explosion!



Fin