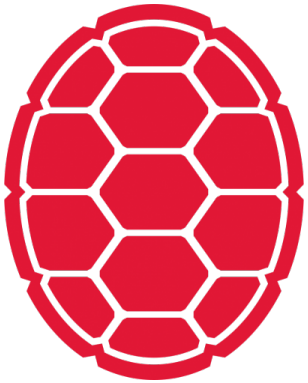


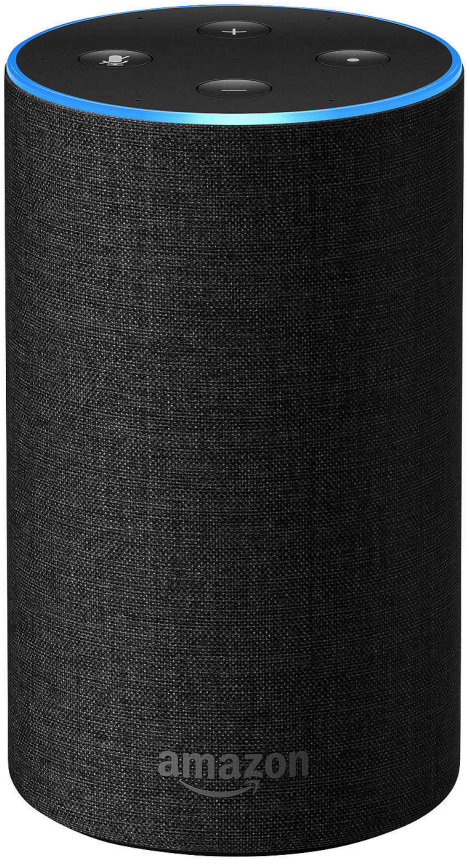
Computational Linguistics Introduction

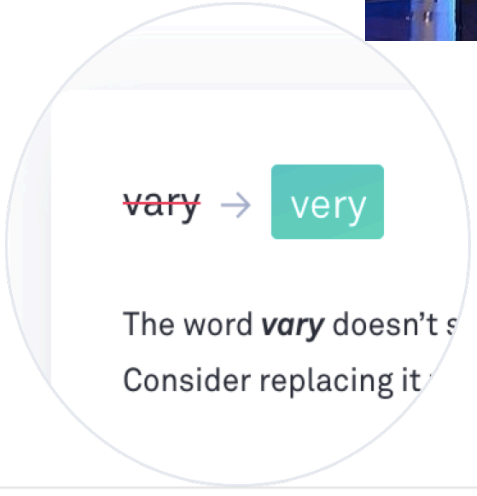
Computational Linguistics: Jordan Boyd-Graber
University of Maryland
00A

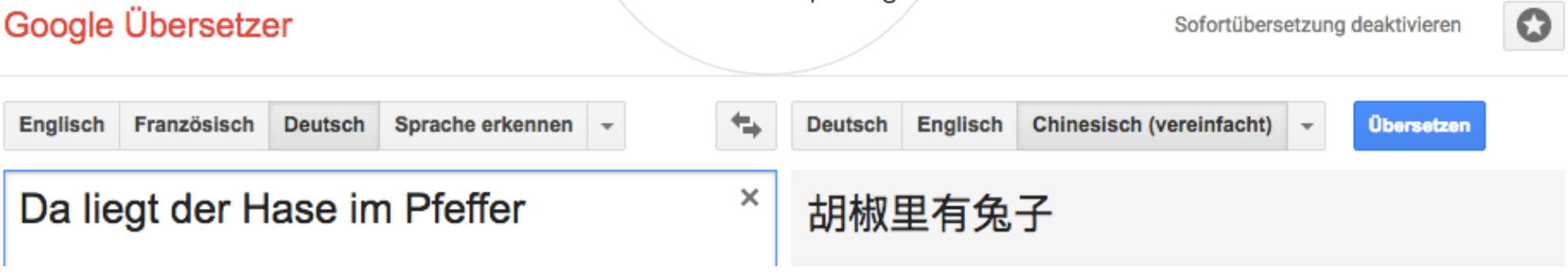
Roadmap

By the end of this class you should . . .

- Be able to give examples of where NLP is used
- Understand the workings of the course
- Know me and each other a little better







Computational Linguistics Applications

Machine Learning is Doing Great!

- Can drive a million miles without an accident
- Can beat any living chess player

Machine Learning is Doing Great!

- Can drive a million miles without an accident
- Can beat any living chess player
- Automated call center vs. five-year old?

Machine Learning is Doing Great!

- Can drive a million miles without an accident
- Can beat any living chess player
- Automated call center vs. five-year old?
- We'll learn why we're so far away

What's Natural Language Processing

- Computational approaches to understand, generate, and process natural language
- Cross-discipline
  - Computer science: implement algorithms
  - Linguistics: develop theory / data
  - Statistics: learn patterns from data
  - Experts in specific languages: get a computer to handle a new language
  - Psychologists: how does our brain process language
  - Sociologists: how do social constraints change how we process language

What computational linguistics can do!

Automatic solutions to . . .

- Explain why the “ly” in “al**ly**” and “quick**ly**” are different (morphology)
- Tell the difference in category between “**water** the flowers” and “drink the **water**” (part of speech tagging)
- Why “saw the sun with the telescope” is different from “saw the astronomer with the telescope”
- Translate “My hovercraft is full of eels” into Hungarian (machine translation)

What you need for this course

- Interactive classroom: come ready with questions and participate in exercises
- Helps to have a laptop to bring to class
- Math background
  - Will ask you to manipulate equations
  - Will expect you to be able to do basic derivatives
  - Work with functions like exponentiation and logs
  - Probability: review online (hugely important)
- Computer / programming skills
  - You will need to write python programs
  - You will need to interact with a Unix command line
  - You will need to interact with data files

Administrivia

- Sign up on Piazza (use a photo)
  - Another way to earn participation
  - caete: only way to earn participation
- Keep track of course webpage `http://umiacs.umd.edu/~jbg/teaching/CMSC_723/`
- Homeworks
- 7 late days
- Let me know about special needs

Course reading

- We will provide reading assignments, mostly from the book. (Read them **before** associated class.)
- The reading will cover more than we cover in class.
- Free online

Communicating with Piazza

We will use Piazza to manage all communication

`https://piazza.com/umd/fall2018/cmsc723/home`

- Questions answered within 1 day (hopefully sooner)
- Hosts discussions among yourselves
- Use for any kind of technical question
- Use for **most** administrative questions
- Can use to send us private questions too

How to ask for help

- Explain what you're trying to do
- Give a minimal example
  - Someone else should be able to replicate the problem easily
  - Shouldn't require any data / information that only you have
- Explain what you **think** should happen
- Explain what you get instead (copy / paste or screenshot if you can)
- Explain what else you've tried

Me

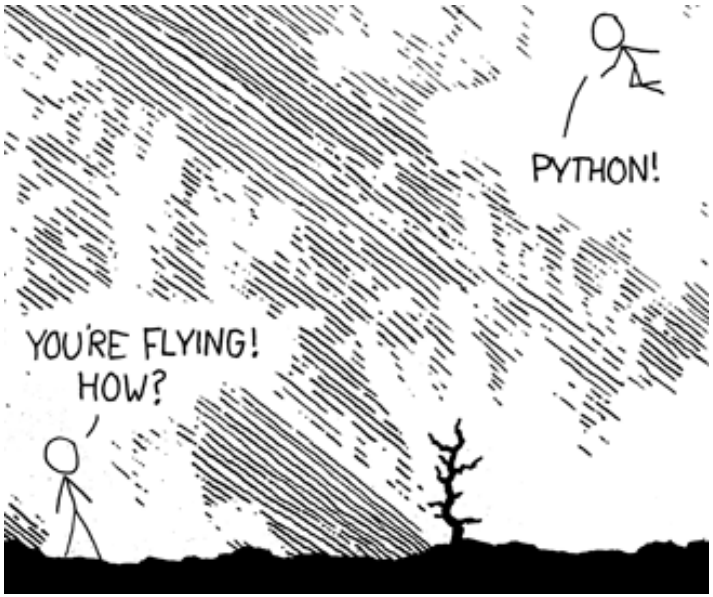
- Associate professor
  - Office: 111B ECCS
- Was formerly a professor at University of Colorado
- Research: topic models, question answering, machine translation
- Fourth time teaching the class (or something similar)
- Born in Colorado (where all my family live)
- Grew up in Iowa (hometown: Keokuk, Iowa)
- Went to high school in Arkansas
- Undergrad in California
- Grad school in New Jersey
- Brief jobs in between:
  - Working on electronic dictionary in Berlin
  - Worked on Google Books in New York
- ying / jbg / jordan / boyd-graber

TAs

- Chen Zhang
- Ahmed Elgohary
- Office hours scheduled soon

Why Python?

- Easy to learn
- Widespread
- Can be fast if you need it (cython)



Why NLTK?

- Handy code for accessing data
- Implementations of standard algorithms
- Easy to quickly process text and try things out

Why NLTK?

- Handy code for accessing data
- Implementations of standard algorithms
- Easy to quickly process text and try things out
- Chapter 1 of NLTK book
- Ask questions on Piazza

Before class . . .

- Install nltk on your computer
- Come to class ready to work through some Python examples
- Also look at homework 1 and see if you have any questions