CS388: Natural Language Processing

Lecture 1: Introduction

Greg Durrett
(he/him)

TEXAS
The University of Texas at Austin

Administrivia

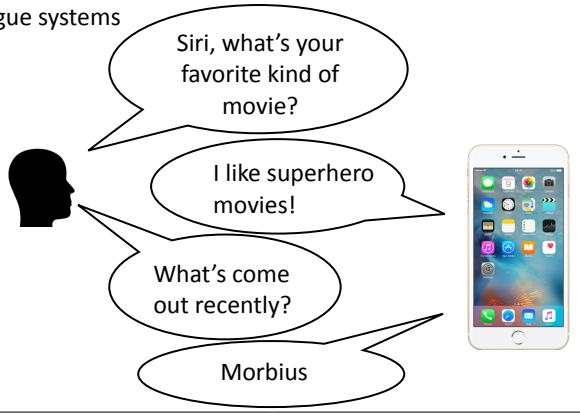
- Lecture: Tuesdays and Thursdays 12:30pm - 1:45pm; recordings made available
- Course website: http://www.cs.utexas.edu/~gdurrett/courses/sp2024/cs388.shtml
- Gradescope: linked from Canvas
- EdStem: linked from Canvas
- TA: Anisha Gunjal
- See course website for OHs

Course Requirements

- 391L Machine Learning (or equivalent)
- 311 or 311H Discrete Math for Computer Science (or equivalent)
- Python experience
- Additional prior exposure to probability, linear algebra, optimization, linguistics, and NLP useful but not required
- Project 1 is out now — take a look at it soon if you have any doubts about the class (we will move quickly through basic classification and neural networks)

What's the goal of NLP?

- Be able to solve problems that require deep understanding of text
- Example: dialogue systems



Machine Translation

The Political Bureau of the CPC Central Committee July 30 hold a meeting

中共中央政治局7月30日召开会议，会议分析研究当前经济形势，部署下半年经济工作。

People's Daily, August 10, 2020

Translate

The Political Bureau of the CPC Central Committee held a meeting on July 30 to analyze and study the current economic situation and plan economic work in the second half of the year.

Question Answering

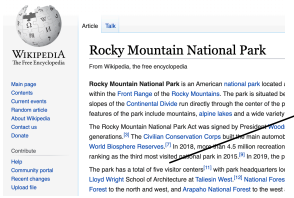
When was Abraham Lincoln born?

map to `Birthday` field

Name	Birthday
Lincoln, Abraham	2/12/1809
Washington, George	2/22/1732
Adams, John	10/30/1735

→ **February 12, 1809**

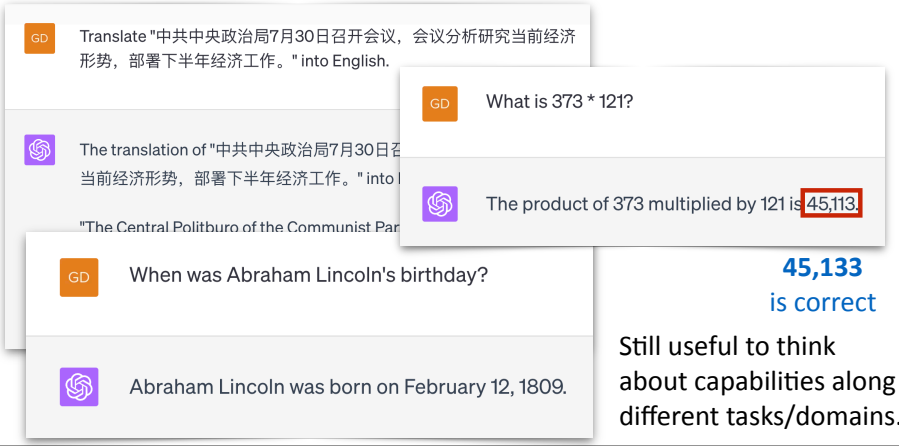
How many visitors centers are there in Rocky Mountain National Park?



The park has a total of five visitor centers

five

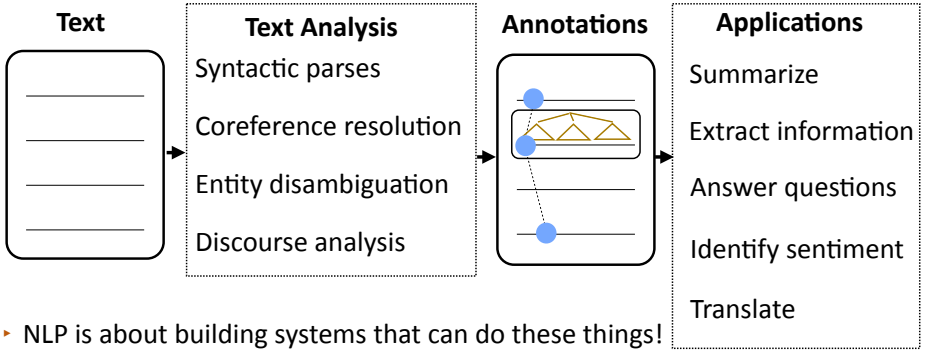
Generalist Systems?



45,133 is correct

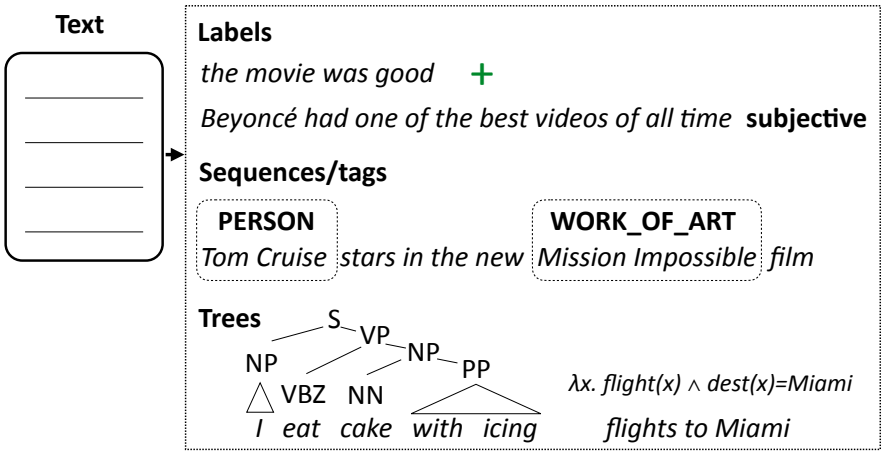
Still useful to think about capabilities along different tasks/domains.

Classic NLP Analysis Pipeline

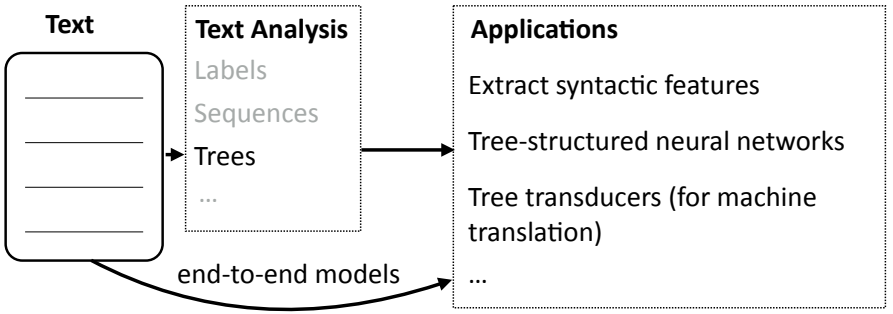


- NLP is about building systems that can do these things!
- All of these components are modeled with statistical approaches trained with machine learning

How do we represent language?



How do we use these representations?



- Both classical pipeline NLP systems and LLM systems like ChatGPT still need to resolve the same issues
- Main question: What ambiguities do we need to resolve in language?

Why is language hard?

(and how can we handle that?)

Language is Ambiguous!

- Hector Levesque (2011): "Winograd schema challenge" (named after Terry Winograd, the creator of SHRDLU)

The city council refused the demonstrators a permit because they advocated violence

The city council refused the demonstrators a permit because they feared violence

The city council refused the demonstrators a permit because they _____ violence

- >5 datasets in the last few years examining this problem and commonsense reasoning
- Referential ambiguity

Language is Ambiguous!

Teacher Strikes Idle Kids

Ban on Nude Dancing on Governor's Desk

Iraqi Head Seeks Arms

- Syntactic and semantic ambiguities: parsing needed to resolve these, but need context to figure out which parse is correct

example credit: Dan Klein

Language is **Really** Ambiguous!

- There aren't just one or two possibilities which are resolved pragmatically

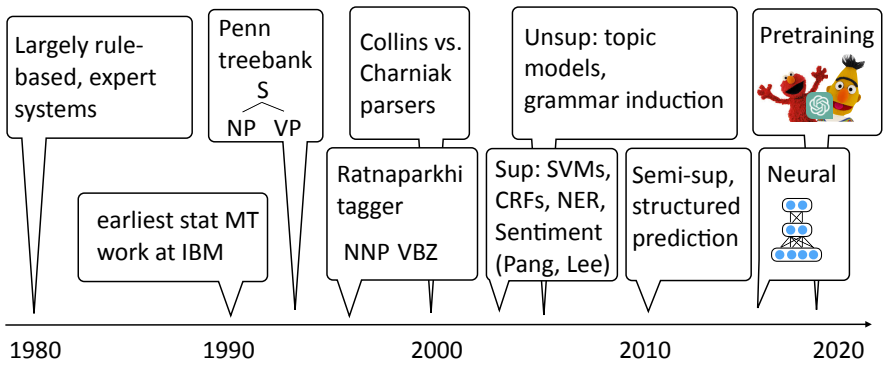
il fait vraiment beau →
- It is really nice out
- It's really nice
- The weather is beautiful
- It is really beautiful outside
- He makes truly beautiful
- It fact actually handsome

- Combinatorially many possibilities, many you won't even register as ambiguities, but systems still have to resolve them

What techniques do we use?

(to combine data, knowledge, linguistics, etc.)

A brief history of (modern) NLP



Pretraining

- Language modeling: predict the next word in a text $P(w_i|w_1, \ldots, w_{i-1})$

P(*w* | I want to go to) = 0.01 Hawai'i
0.005 LA
0.0001 class

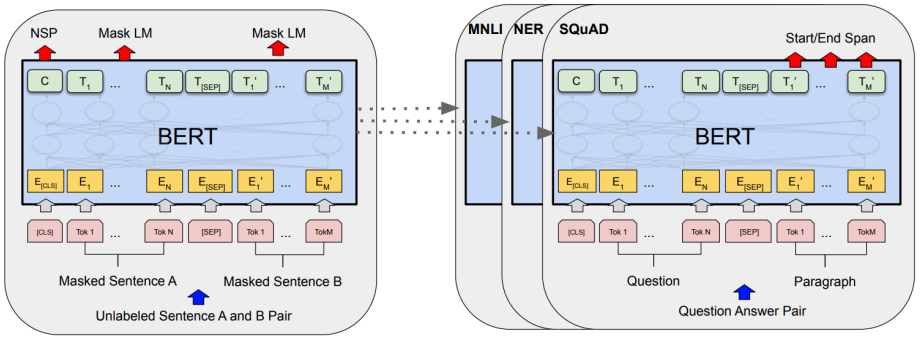
: use this model for other purposes

P(*w* | the acting was horrible, I think the movie was) = 0.1 bad
0.001 good

- Model understands some sentiment?
- Train a neural network to do language modeling on massive unlabeled text, fine-tune it to do {tagging, sentiment, question answering, …}

Peters et al. (2018), Devlin et al. (2019)

BERT



Pre-training Fine-Tuning

- Key parts which we will study: (1) Transformer architecture; (2) what data is used (both for pre-training and fine-tuning)

Devlin et al. (2019)

GPT and In-Context Learning

- Even more "extreme" setting: no gradient updates to model, instead large language models "learn" from examples in their context
- Many papers studying why this works. We will read some!

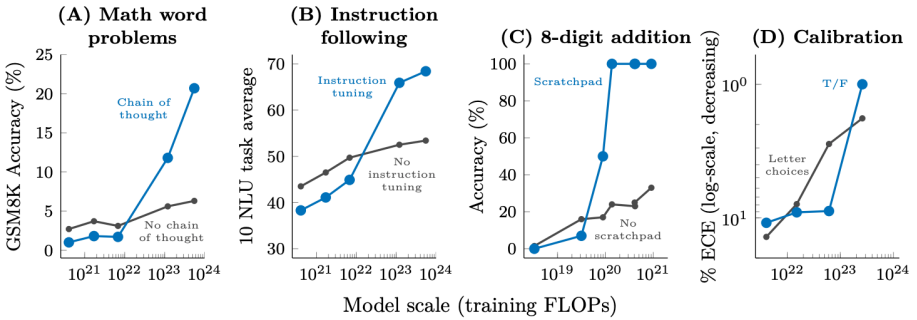
Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

```
1  Translate English to French:        ← task description
2  sea otter => loutre de mer          ← examples
3  peppermint => menthe poivrée
4  plush girafe => girafe peluche
5  cheese =>                           ← prompt
```

Brown et al. (2020)

Scaling Laws



- Many of the ideas that are successful in 2024 only make sense and only work because the models are so big!

Kaplan et al. (2020), Jason Wei et al. (2022)

Interpretability

- When we have complex models, how do we understand their decisions?

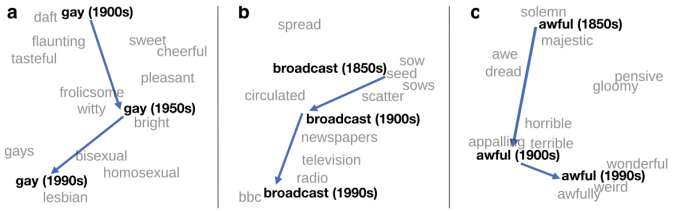
The movie is mediocre, maybe even bad.	**Negative** 99.8%
The movie is mediocre, maybe even ~~bad~~.	**Negative** 98.0%
The movie is ~~mediocre~~, maybe even bad.	**Negative** 98.7%
The movie is ~~mediocre~~, maybe even ~~bad~~.	**Positive** 63.4%
The movie is ~~mediocre~~, ~~maybe~~ even ~~bad~~.	**Positive** 74.5%
The ~~movie~~ is mediocre, maybe even ~~bad~~.	**Negative** 97.9%

The movie is mediocre, maybe even bad.

Wallace, Gardner, Singh
Interpretability Tutorial at EMNLP 2020

NLP vs. Computational Linguistics

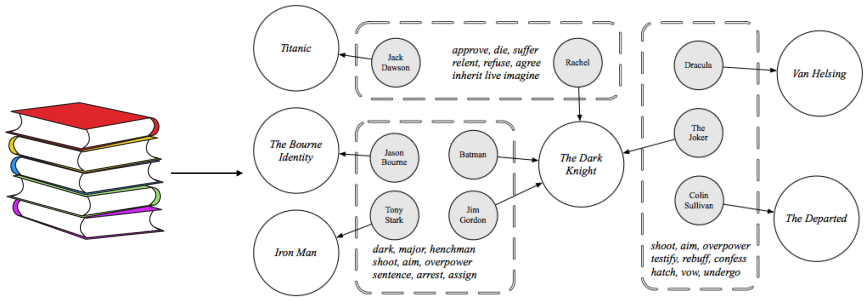
- NLP: build systems that deal with language data
- CL: use computational tools to study language



Hamilton et al. (2016)

NLP vs. Computational Linguistics

- Computational tools for other purposes: literary theory, political science…



Bamman, O'Connor, Smith (2013)

Where are we?

- We have very powerful neural models that can fit lots of datasets
- Data: we need data that is not just correctly labeled, but reflects what we actually want to be able to do
- Users: systems are not useful unless they do something we want
- Language/outreach: who are we building this for? What languages/dialects do they speak?

Social Impact/Ethics

- NLP systems are increasingly used in the world

…and increasingly we have to reckon with their impact

- This lecture: let's warm up by thinking about these issues a bit

Social Impact/Ethics

- What is one scenario where you think deployment of an NLP system might pose ethical challenges *due to the application* itself (i.e., using NLP to do "bad stuff")?

- What is a scenario where you think deployment of an NLP system might pose ethical challenges due to *unintended* consequences (e.g., unfairness, indirectly causing bad things to happen, etc.).

Syllabus

Outline

- Classification: linear and neural, word representations (2 weeks)
- Language modeling, transformers, and pre-training (2 weeks)
- Dataset biases, interpretability, rationales, advanced pre-training (3 weeks)
- Structured prediction, tagging, parsing (1.5 weeks)
- Applications and misc (3 weeks)

Course Goals

- Cover fundamental machine learning and deep learning techniques used in NLP
- Understand how to look at language data and approach linguistic phenomena
- Cover modern NLP problems encountered in the literature: what are the active research topics in 2024?
- Make you a "producer" rather than a "consumer" of NLP tools
  - The assignments should teach you what you need to know to understand nearly any system in the literature (classification layers from Project 1, Transformer backbones from Project 2, datasets and what gets learned from Project 3)

Assignments

- Three projects (10%/15%/20%)
  - Implementation-oriented, with an open-ended component to each
  - Project 1 (linear and neural classification) is out NOW
  - ~2 weeks per project, 5 "slip days" for automatic extensions
- Projects are graded on a mix of code performance, writeup, and "extensions" that you explore on top of what's required

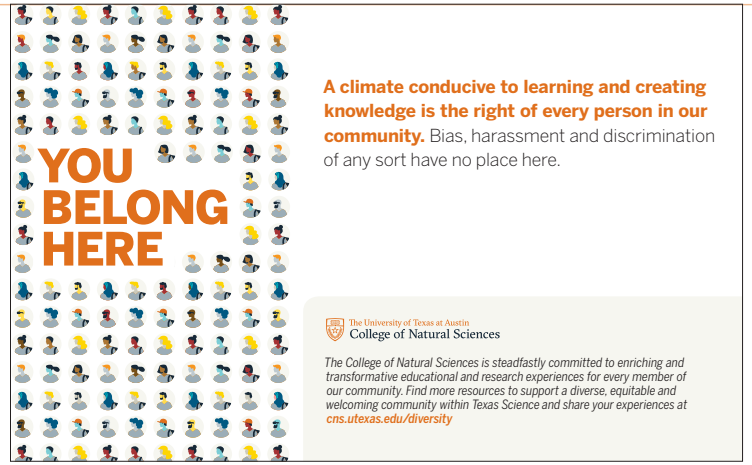
These projects require understanding of the concepts, ability to write performant code, and ability to think about how to debug complex systems. **They are challenging, so start early!**

See the syllabus for details on usage of ChatGPT and Copilot. You can use these, but you must disclose them and you should not rely on ChatGPT to generate much text (it will often produce poor text)

Assignments

- Final project (55%)
  - Groups of 2 preferred, 1 is possible
  - (Brief!) proposal to be approved by course staff by the midpoint of the semester
  - Written in the style and tone of a CS conference paper
- Compute:
  - Google Colab is a nice resource for projects (especially Colab Pro, $9.99/mo)
  - Unfortunately, we cannot provide OpenAI / etc. credits
  - When you propose projects, we will discuss feasibility given your compute resources available

Conduct



Survey (on Instapoll)