# COS 484: Natural Language Processing

## L1: Introduction + Language Models

Spring 2026

# Course staff

## Instructors

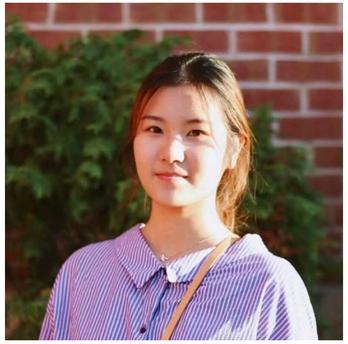
Tri Dao

Karthik Narasimhan

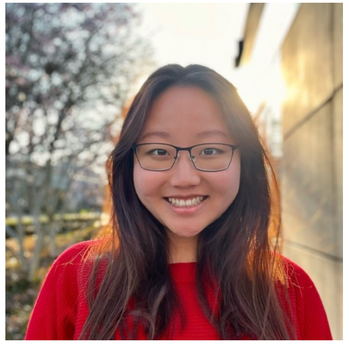
## TAs

Max Gupta

Lucy He



Sijia Liu



Ambri Ma

Keerthana Nallamotu

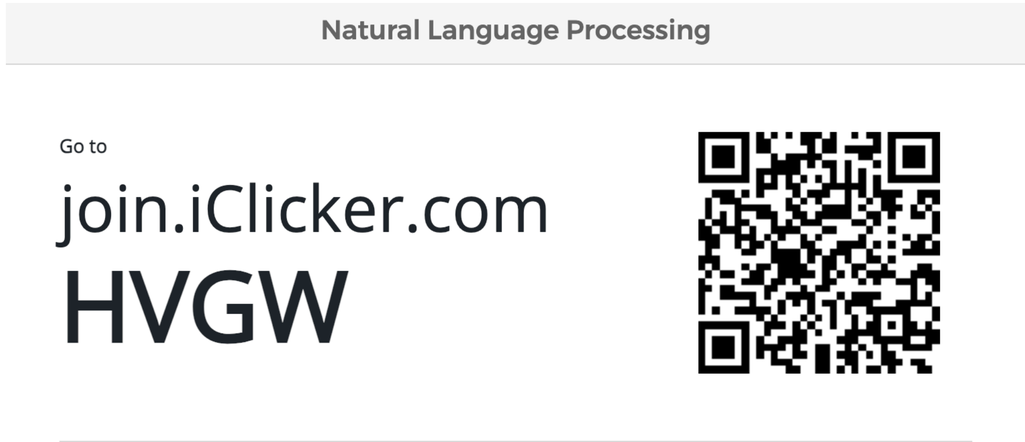
Howard Yen

# Logistics

**Course webpage**: https://nlp.cs.princeton.edu/cos484/

- Contains all information on the course
- Ed will be used for all announcements - make sure you have **notifications** turned on!
  - No emails
  - You can use private posts to course staff for sensitive matters
  - Or come to office hours!
- We'll use iClicker for in class polls

- **Precepts:** 1-hour precept every week taught by TAs (optional)
  - Fridays, 12-1pm



# Assignments

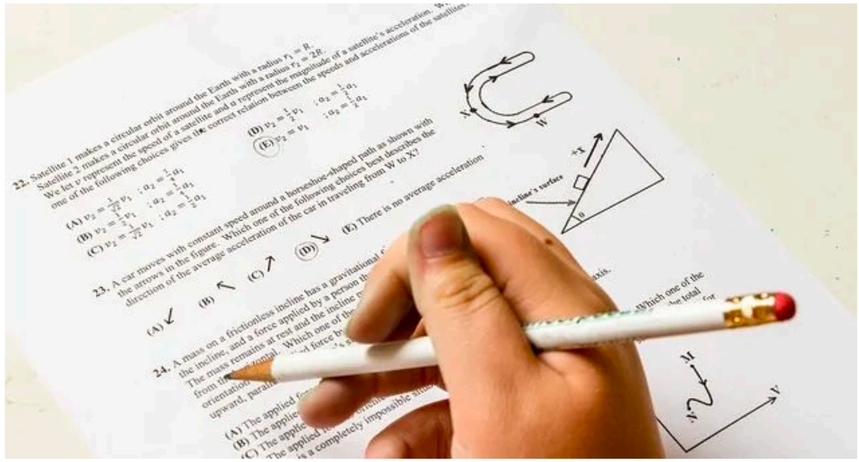
**Assignments (40%)**: 4 total

- **A1, A2, A3, A4**: 10% each - each assignment has 2 or 3 weeks
- Every assignment has a **theory** component and a **programming** component
- You cannot use AI tools for theory parts of assignments
- You may use AI tools for programming parts, but need to disclose.

You have **4 free late days** for all assignments; After that, 10% penalty for each late day (up to a maximum of 3 days beyond which submissions will not be accepted) - see website for details

# Midterm

- **Midterm (25%)**
  - In person, March 5 (more details will be announced soon)
  - All the topics up to and including Feb 26 will be covered
  - No final exam



# Final Project

**Final project (35%)**

- Complete in a team of 3
- Open-ended research project
- Proposal (0%) due before - date will be announced soon

**Extra bonus (5%)** - participation in class and Ed discussions

*No **pre-determined cut-offs for final grades**, will be decided at the end taking into account the performance of the entire class and will be fairly assigned to measure your level of understanding of the subject.*

# Textbooks

(NLP is a rapid-moving field…)

https://web.stanford.edu/~jurafsky/slp3/

**Speech and Language Processing** (3rd ed. draft)

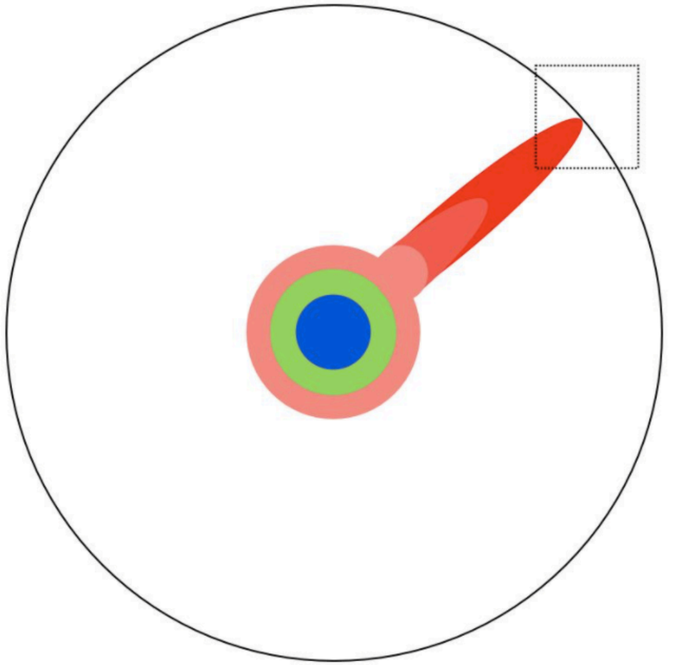
**Dan Jurafsky and James H. Martin**


**Here's our Jan 7, 2023 draft! This draft is mostly a bug-fixing and restructuring release, there are no r**
**the applications section earlier, reflecting how we and others tend to teach NLP, and combines the linguisti**

A good way to learn about state-of-the-art NLP concepts is through **research papers** and **blog posts**

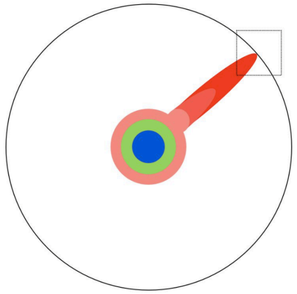
# Course goals

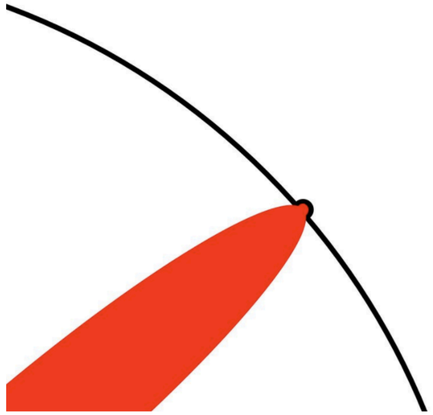
- Learn theoretical fundamentals
- Gain practical experience with modern NLP tools
- Carry out independent exploration



# This is an advanced class

Image: https://matt.might.net/articles/phd-school-in-pictures/





# This is a *very* advanced class

Image: https://matt.might.net/articles/phd-school-in-pictures/

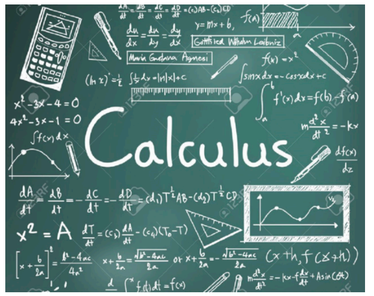
# Prerequisites

- **Required: COS324**, knowledge of probability, linear algebra, calculus
- Be ready to pick up new ML concepts
- Proficiency in Python: programming assignments and projects will require use of Python, Numpy and PyTorch.

Q. Why is COS324 a prerequisite?

We assume you have learned the following concepts already:

- Language models
- Logistic regression w/ regularization
- Unsupervised vs supervised learning
- Feedforward neural networks, convolutional neural networks
- PyTorch programming
- (A little bit of reinforcement learning)




PyTorch

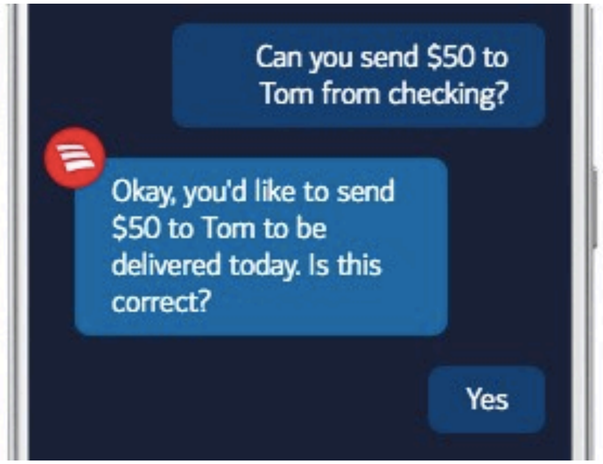
# Natural Language Processing

- NLP = building **computer programs** to analyze, understand and generate **human language - either spoken or written** (informal)
- NLP is an interdisciplinary field


# Natural Language Processing

- NLP = building **computer programs** to analyze, understand and generate **human language - either spoken or written** (informal)

Communication with humans (ex. personal assistants, customer service)



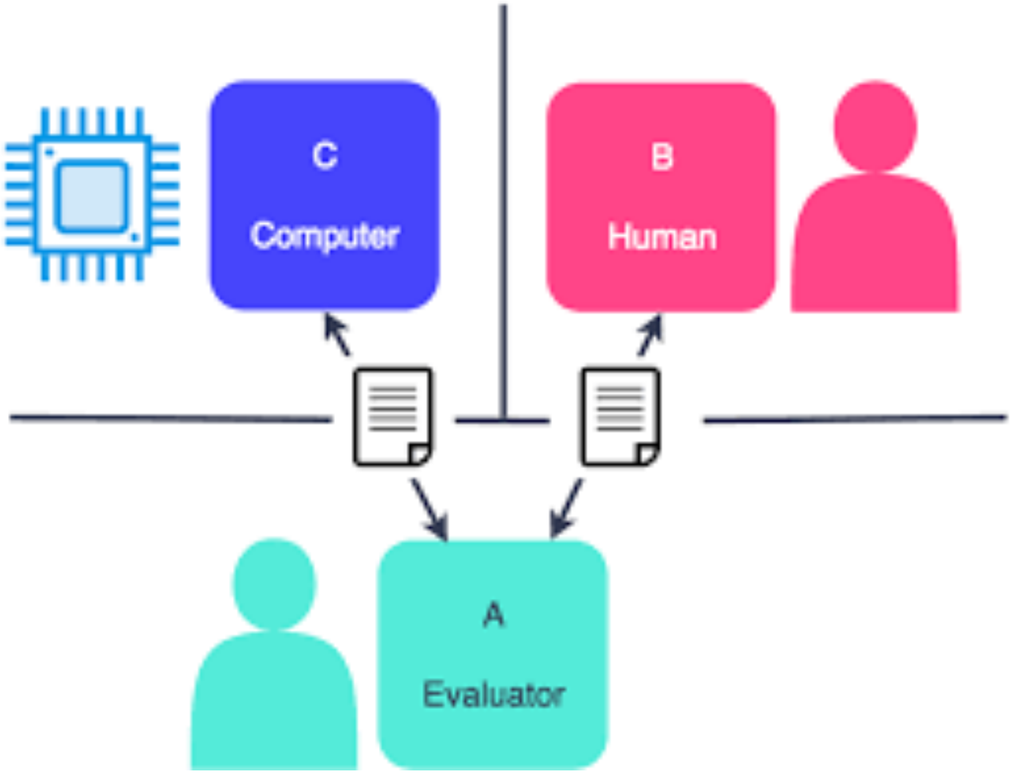

Banking assistant

ONLINE

Access the wealth of information about the world — crucial for AI systems

OFFLINE

# Turing Test

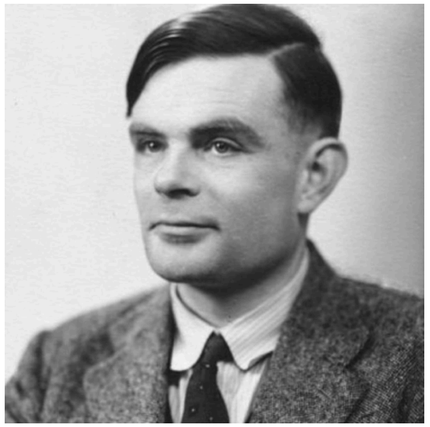



A. M. Turing (1950) Computing Machinery and Intelligence. *Mind 49:* 433-460.

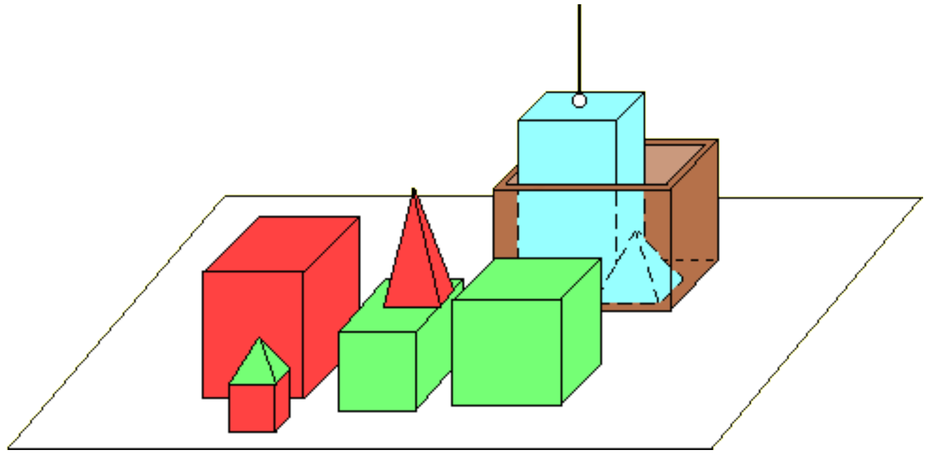
**COMPUTING MACHINERY AND INTELLIGENCE**

**By A. M. Turing**

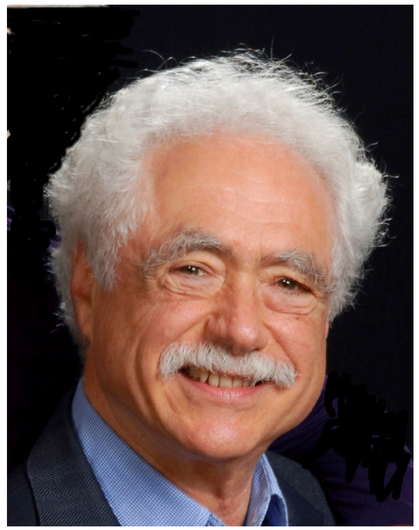
**1. The Imitation Game**



Ability to understand and generate language ~ intelligence



SHRDLU,
1968



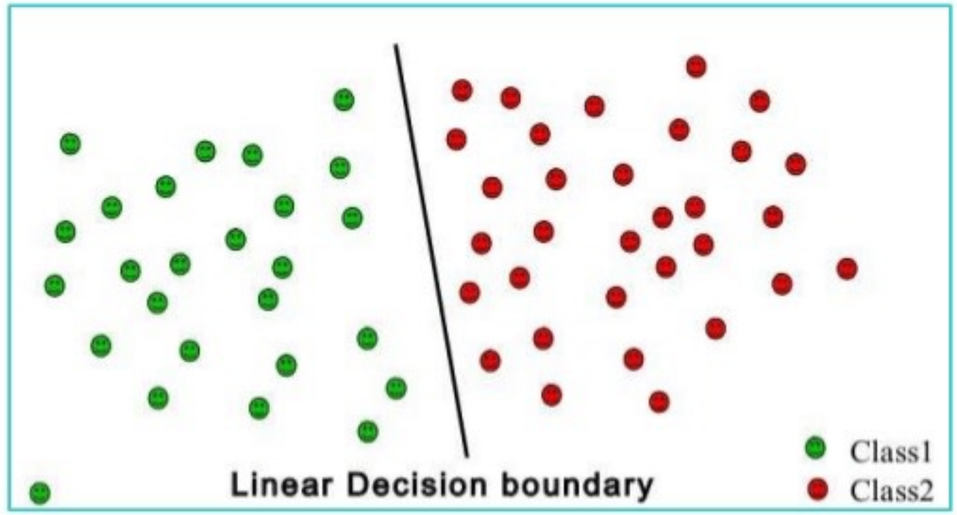
Terry Winograd

> How many red blocks are there?
- THREE OF THEM

> Pick up the red block on top of a green one
OK.

- Rule-based, requiring extensive programming
- Limited domain

# Statistical learning

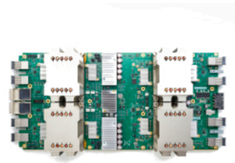



- Use of machine learning techniques in NLP
- Increase in computational capabilities
- Availability of electronic corpora

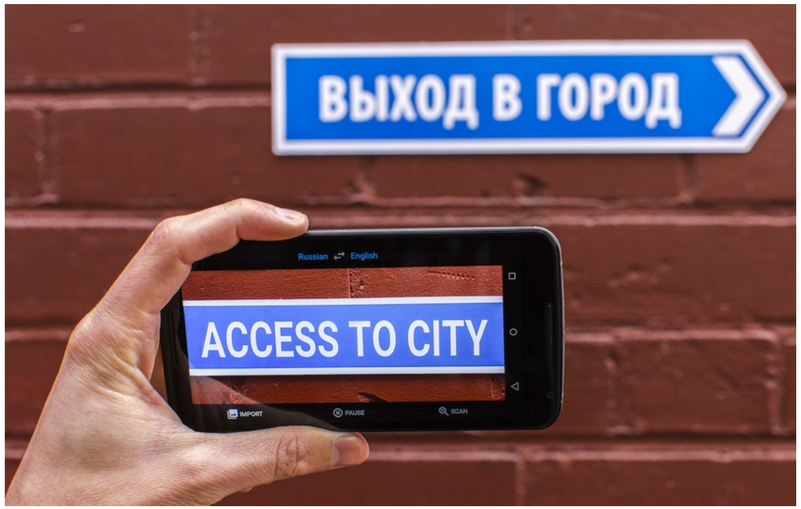
# The era of deep learning

- Significant advances in core NLP technologies
- Essential ingredient: large-scale supervision, lots of compute
- Reduced manual effort - less/zero **feature engineering**


GPU



TPU





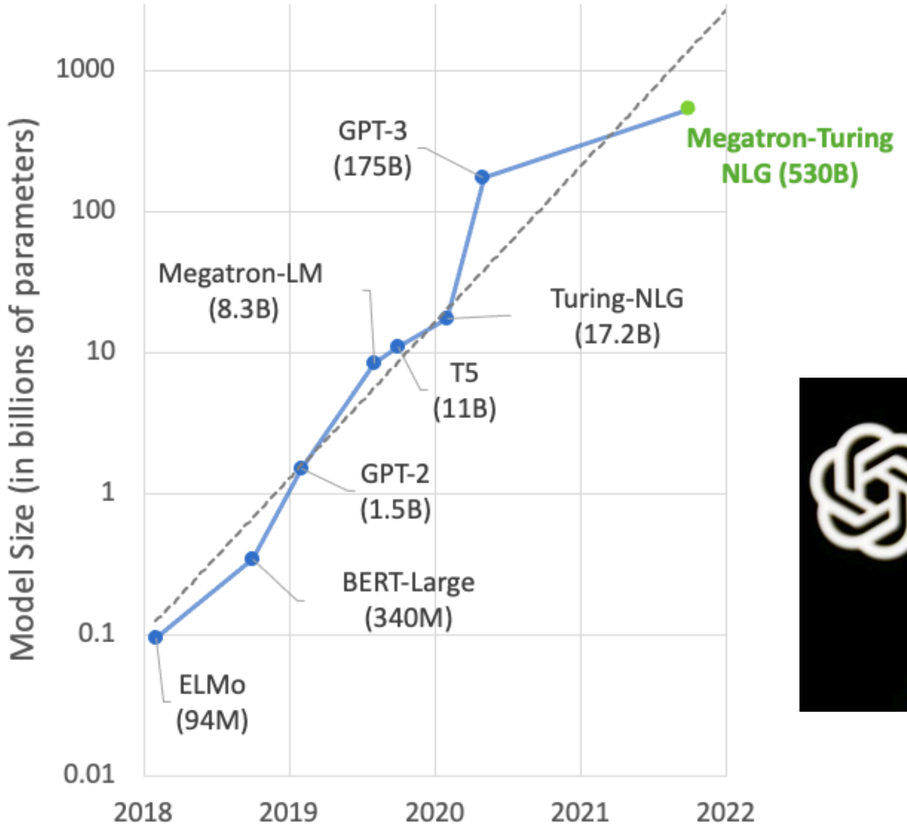
36M sentence pairs

*Russian*: Машинный перевод - это круто!

*English:* Machine translation is cool!

# The scaling years

BERT, ELMo, ERNIE…





- Leverages a lot of unlabeled text
- Model size increased by $10^3 - 10^5$x in parameters

# Why is language difficult to understand?

# Why is language difficult to understand?

- Ambiguous
- Dialects
- Accents
- listener has to infer - pragmatics
- humor, sarcasm, irony
- context, dependencies

# Lexical ambiguity

The fisherman went to the **bank**.

**bank**[1]

/baNGk/

*noun*

plural noun: **banks**

1. the land alongside or sloping down to a river or lake.
   "willows lined the bank"
   *synonyms:* edge, side, shore, coast, embankment, bankside, levee, border, verge, boundary, margin, rim, fringe; More

1. a financial establishment that invests money deposited by customers, pays it out when required, makes loans at interest, and exchanges currency.
   "I paid the money straight into my bank"
   *synonyms:* financial institution, merchant bank, savings bank, finance company, trust company,

One word can mean several different things

# Lexical ambiguity

The fisherman went to the **bank**. He deposited some money.

**bank**[1]

/baNGk/

*noun*

plural noun: **banks**

1. the land alongside or sloping down to a river or lake.
   "willows lined the bank"
   *synonyms:* edge, side, shore, coast, embankment, bankside, levee, border, verge, boundary, margin, rim, fringe; More

1. a financial establishment that invests money deposited by customers, pays it out when required, makes loans at interest, and exchanges currency.
   "I paid the money straight into my bank"
   *synonyms:* financial institution, merchant bank, savings bank, finance company, trust company,

Word sense disambiguation

# Lexical variations

**ACCORDING TO THE THESAURUS, "THEY'RE HUMID, PREPOSSESSING HOMOSAPIENS WITH FULL SIZED AORTIC PUMPS" MEANS "THEY'RE WARM, NICE PEOPLE WITH BIG HEARTS."**

Several words can mean the same thing!

# Comprehending word sequences

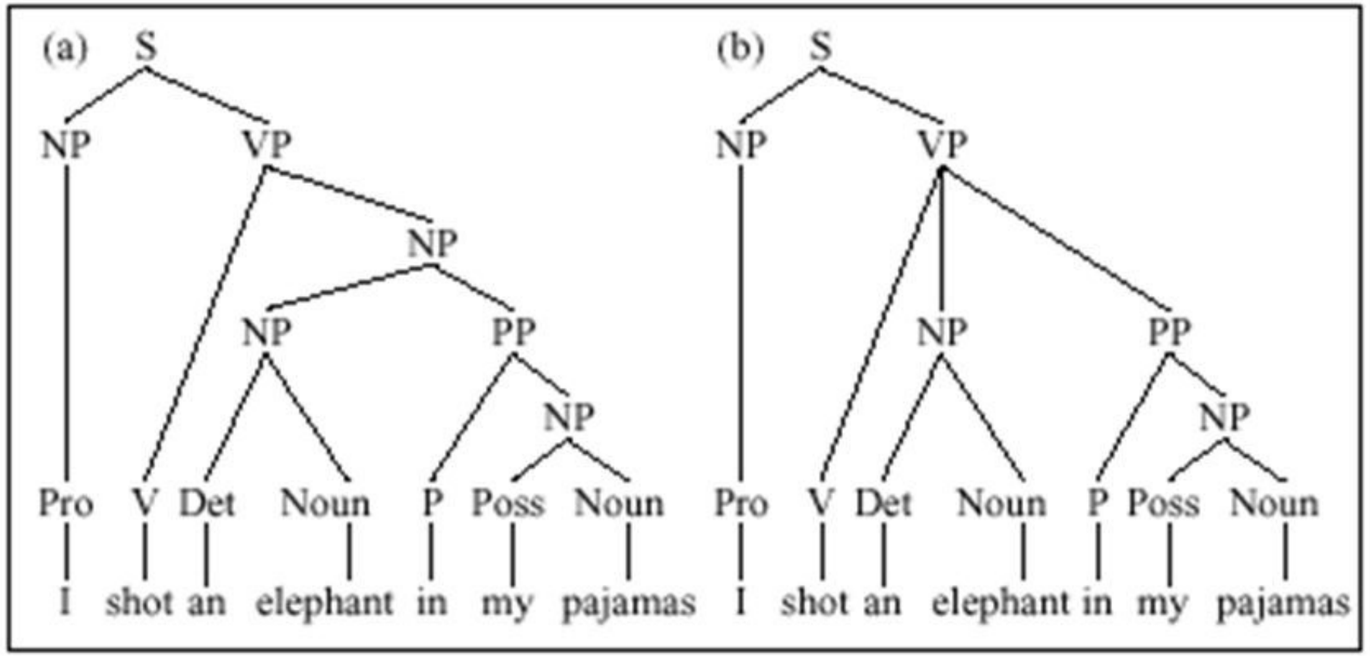
- My brother went to the park near my sister's house
- Park my went house near to sister's my brother the
- "My brother went park near sister's house"?
- The old man the boat
- The cotton clothing is made of grows in Mississippi

Garden Path sentence

- Implicit structure in all languages
- Coarse-to-fine levels (recursive)
- What are some good data structures to represent this?

# Syntactic ambiguity

I shot an elephant in my pajamas




Human language is full of such examples!

# Discourse ambiguity

- The man couldn't lift his son because he was so heavy.
- The man couldn't lift his son because he was so weak.

What does "he" refer to?

- The city councilmen refused the demonstrators a permit because **they** feared violence.
- The city councilmen refused the demonstrators a permit because **they** advocated violence.

What does "they" refer to?

Anaphora resolution

# Roadmap

1

- N-gram language models
- Text classification
- Word embeddings
- Neural nets for language

2

- Sequence models
- Recurrent neural nets
- Attention
- Transformers

3

- Large language models
  - Pre-training
  - Adaptation
  - Post-training
- Language Agents

4

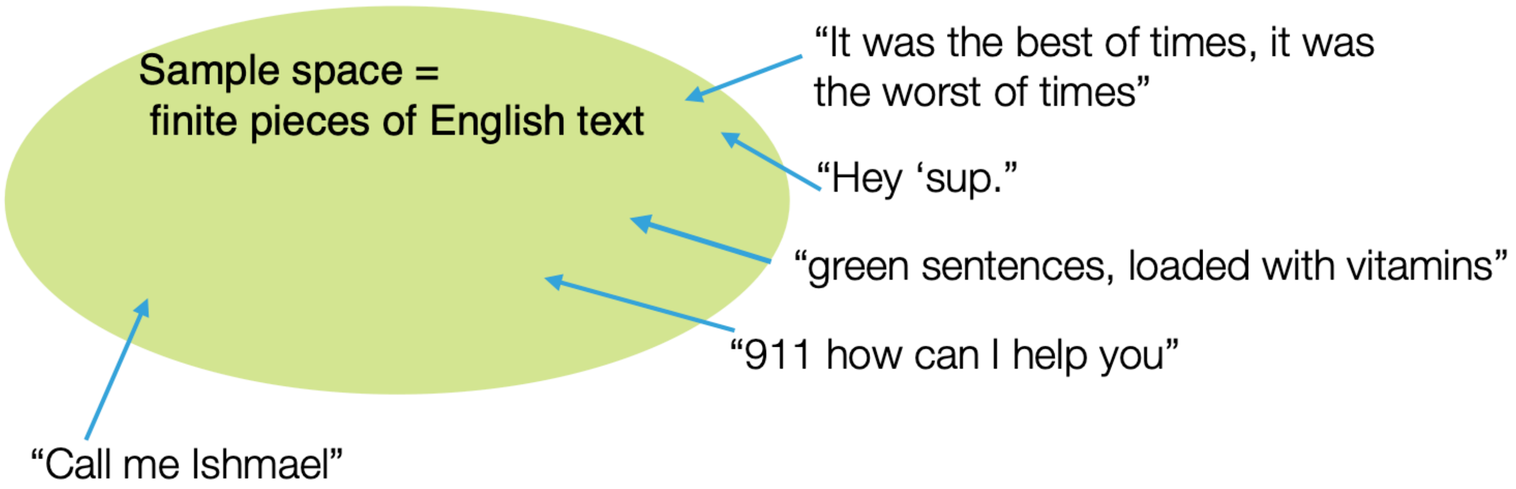
- Systems for LLMs
- Evals + data
- Reasoning

# Language models

# What is a language model?

- A probabilistic model of a sequence of words
- Joint probability distribution of words $w_1, w_2, \ldots, w_n$:

$$P(w_1, w_2, w_3, ..., w_n)$$




**How likely is a given phrase, sentence, paragraph or even an entire document?**

# Chain rule

Conditional probability:
$p(w \mid w_1, w_2), \forall w \in V$

$$p(w_1, w_2, w_3, \ldots, w_n) =$$

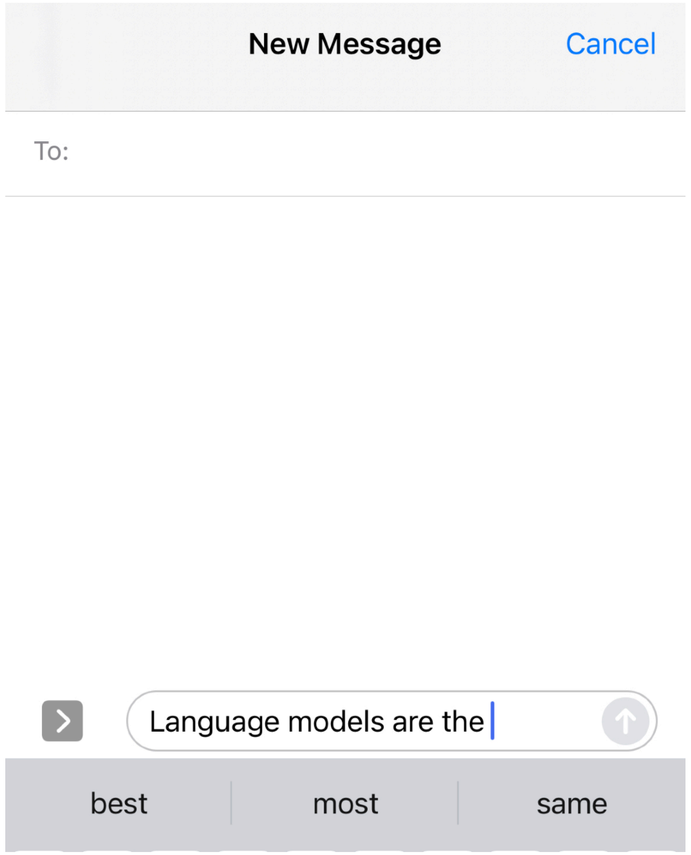
$$p(w_1)p(w_2 \mid w_1)p(w_3 \mid w_1, w_2) \times \cdots \times p(w_n \mid w_1, w_2, \ldots, w_{n-1})$$

Sentence: “the cat sat on the mat”

$$P(\text{the cat sat on the mat}) = P(\text{the}) * P(\text{cat}|\text{the}) * P(\text{sat}|\text{the cat})$$
$$*P(\text{on}|\text{the cat sat}) * P(\text{the}|\text{the cat sat on})$$
$$*P(\text{mat}|\text{the cat sat on the})$$

Implicit order

# Language models are everywhere

# Estimating probabilities

$$P(\text{sat}|\text{the cat}) = \frac{\text{count}(\text{the cat sat})}{\text{count}(\text{the cat})}$$

$$P(\text{on}|\text{the cat sat}) = \frac{\text{count}(\text{the cat sat on})}{\text{count}(\text{the cat sat})}$$

$$\vdots$$

Maximum likelihood estimate (MLE)

Assume we have a vocabulary of size $V$,

how many sequences of length $n$ do we have?

A) $n * V$

B) $n^V$

C) $V^n$

D) $V/n$

Natural Language Processing

Go to

join.iClicker.com

HVGW

# Estimating probabilities

$$P(\text{sat}|\text{the cat}) = \frac{\text{count}(\text{the cat sat})}{\text{count}(\text{the cat})}$$

$$P(\text{on}|\text{the cat sat}) = \frac{\text{count}(\text{the cat sat on})}{\text{count}(\text{the cat sat})}$$

$$\vdots$$

Maximum likelihood estimate (MLE)

- With a vocabulary of size V, # sequences of length n = $V^n$
- Typical English vocabulary ~ 40k words
  - Even sentences of length $\leq 11$ results in more than $4 \times 10^{50}$ sequences. Too many to count!
  - *(For reference, # of atoms in the earth* $\sim 10^{50}$*)*

# Markov assumption

- Use only the recent past to predict the next word
- Reduces the number of estimated parameters in exchange for modeling capacity
- 1st order

$$P(\text{mat}|\text{the cat sat on the}) \approx P(\text{mat}|\text{the})$$

- 2nd order

$$P(\text{mat}|\text{the cat sat on the}) \approx P(\text{mat}|\text{on the})$$

Andrey Markov

# $k^{th}$ order Markov

Consider only the last *k* words (or less) for context which implies the probability of a sequence is:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

(assume $w_j = \phi \quad \forall j < 0$)

Need to estimate counts for up to (k+1) grams

# n-gram models

Unigram $$P(w_1, w_2, ...w_n) = \prod_{i=1}^{n} P(w_i)$$ e.g. P(the) P(cat) P(sat)

Bigram $$P(w_1, w_2, ...w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1})$$ e.g. P(the) P(cat | the) P(sat | cat)

and Trigram, 4-gram, and so on.

*Larger the n, more accurate and better the language model*
*(but also higher costs)*

*Caveat: Assuming infinite data!*

# Estimating probabilities

Consider the following corpus

<s> I like apples </s>

<s> You like strawberries </s>

<s> You like apples </s>

Note: <s> and </s> are starting and ending tokens

What's the bigram probability P(apples | like) ?

(A) 1/3 (B) 2/3 (C) 1/2 (D) 1

# Estimating probabilities

Consider the following corpus

<s> I like apples </s>

<s> You like strawberries </s>

<s> You like apples </s>

Note: <s> and </s> are starting and ending tokens

What's the bigram probability P(apples | like) ?

(A) 1/3 (B) 2/3 (C) 1/2 (D) 1

$$P(\text{apples} \mid \text{like}) = \frac{\text{Count}(\text{"like apples"})}{\text{Count}(\text{"like"})} = \frac{2}{3}$$

# Estimating probabilities

Consider the following corpus

<s> I like apples </s>

<s> You like strawberries </s>

<s> You like apples </s>

Note: <s> and </s> are starting and ending tokens

Using the bigram model, what's the probability of the sentence "<s> I like strawberries </s>"? Ignore the probability of <s>.

(A) 4/9 (B) 1/3 (C) 2/9 (D) 1/9

# Estimating probabilities

Consider the following corpus

<s> I like apples </s>

<s> You like strawberries </s>

<s> You like apples </s>

Note: <s> and </s> are starting and ending tokens

Using the bigram model, what's the probability of the sentence "<s> I like strawberries </s>"? Ignore the probability of <s>.

(A) 4/9 (B) 1/3 (C) 2/9 (D) 1/9

P(<s> I like strawberries </s>) = $\frac{1}{3} \cdot 1 \cdot \frac{1}{3} \cdot 1$

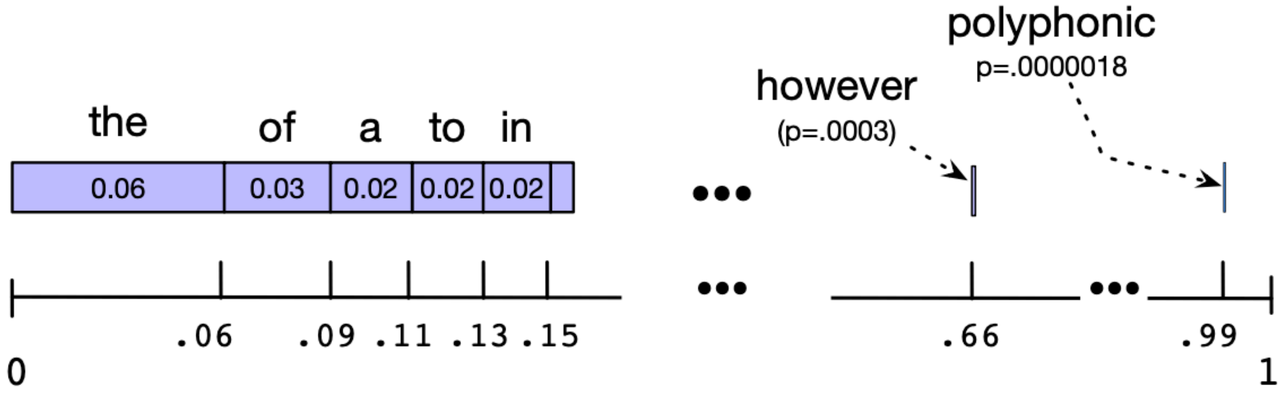
# Generating from a language model

# Generating from a language model

- Given a language model, how to generate a sequence?

$$\text{Bigram} \quad P(w_1, w_2, ...w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1})$$

- Generate the first word $w_1 \sim P(w)$
- Generate the second word $w_2 \sim P(w \mid w_1)$
- Generate the third word $w_3 \sim P(w \mid w_2)$
- …



# Generating from a language model

- Given a language model, how to generate a sequence?

Trigram $$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_{i-2}, w_{i-1})$$

- Generate the first word $w_1 \sim P(w)$
- Generate the second word $w_2 \sim P(w \mid w_1)$
- Generate the third word $w_3 \sim P(w \mid w_1, w_2)$
- Generate the fourth word $w_4 \sim P(w \mid w_2, w_3)$
- …

# Generations

**Unigram**

- *To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have*

**Bigram**

- *Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow.*
- *What means, sir. I confess she? then all sorts, he is trim, captain.*

**Trigram**

- *Fly, and will rid me these news of price. Therefore the sadness of parting, as they say,*
- *This shall forbid it should be branded, if renown made it empty*

Typical n-gram LMs are not sufficient to handle long-range dependencies

"The woman/man could not go to work that day
because she/he had a doctor's appointment"

# Generations

Example from a GPT-2 output (2019):

prompt aka. conditional context

**With the start of the new academic year, Princeton** has an opportunity to help provide a new generation of women with a diverse set of academic resources for higher education.
We are offering the resources of the Princeton-McGill program specifically to women with undergraduate degrees who would like to enhance their academic experience. Princeton-McGill offers a comprehensive suite of services for women and their families including a variety of graduate programs, support programs, and the opportunity to serve as leaders in their communities with a wide variety of programs, activities and services. For the upcoming fall, Princeton-McGill will also offer its Women's Center , which is located in a renovated women's dorm.
At Princeton, we are working with the Princeton-McGill community to develop a suite of programs that are designed to give new and returning students a strong foundation for a successful, rewarding graduate career. The Women's Center , the Princeton-McGill Women's Center provides a range of supports to address the specific needs of female doctoral degree graduates. Programs are tailored to meet the unique needs of women under the age of 28, women and families

https://talktotransformer.com/

$$P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_{i-1024}, \ldots, w_{i-2}, w_{i-1})$$

Modern LMs can handle much longer contexts!

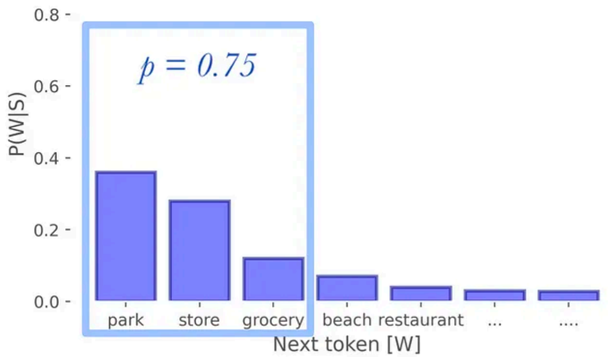
# Generation methods (advanced)

- Greedy: choose the most likely word!
  - To predict the next word given a context of two words $w_1, w_2$:

$$w_3 = \arg\max_{w \in V} P(w \mid w_1, w_2)$$

- Top-k vs top-p sampling: “The boy went to the _____”


Top-k sampling

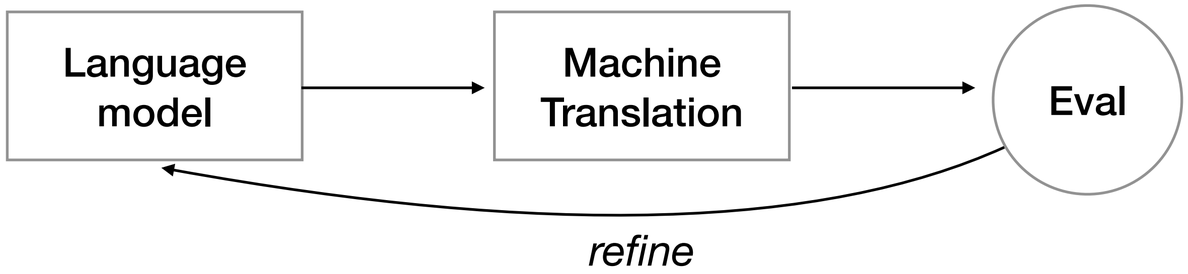



Top-p sampling

https://blog.allenai.org/a-guide-to-language-model-sampling-in-allennlp-3b1239274bc3

# Evaluating a language model

# Extrinsic evaluation




- Train LM → apply to task → observe accuracy
- Directly optimized for downstream applications
  - higher task accuracy → better model
- Expensive, time consuming
- Hard to optimize downstream objective (indirect feedback)

New Approach to Language Modeling Reduces Speech Recognition Errors by Up to 15%

December 13, 2018
Ankur Gandhe

Alexa | Alexa research | Alexa science

# Intrinsic evaluation of language models

- **Train** parameters on a suitable training corpus
  - Assumption: observed sentences ~ good sentences
- **Test** on different, unseen corpus
  - If a language model assigns a higher probability to the test set, it is better
- Evaluation metric - **perplexity**!

# Perplexity (ppl)

- Measure of how well a LM **predicts** the next word
- For a test corpus with words $w_1, w_2, \ldots w_n$

Perplexity = $P(w_1, w_2, \ldots, w_n)^{-1/n}$

$$\text{ppl}(S) = 2^x \quad \text{where} \quad x = -\frac{1}{n}\log_2 P(w_1, \ldots, w_n) = -\frac{1}{n}\sum_{i=1}^{n}\log_2 P(w_i \mid w_1 \ldots w_{i-1})$$

Cross-Entropy

- Unigram model: $x = -\frac{1}{n}\sum_{i=1}^{n}\log P(w_i)$ (since $P(w_j \mid w_1 \ldots w_{j-1}) \approx P(w_j)$ )
- Minimizing perplexity ~ maximizing probability of corpus

# Intuition on perplexity

$$\text{ppl}(S) = 2^x \quad \text{where} \quad x = -\frac{1}{n}\sum_{i=1}^{n} \log P(w_i \mid w_1 \ldots w_{i-1})$$

If our k-gram model (with vocabulary V) has following probability:

$$P(w \mid w_{i-k}, \ldots w_{i-1}) = \frac{1}{|V|}, \qquad \forall w \in V$$

what is the perplexity of the test corpus?

A) $2^{|V|}$ B) $|V|$ C) $|V|^2$ D) $2^{-|V|}$

# Intuition on perplexity

$$\text{ppl}(S) = 2^x \quad \text{where} \quad x = -\frac{1}{n}\sum_{i=1}^{n} \log P(w_i \mid w_1 \ldots w_{i-1})$$

If our k-gram model (with vocabulary V) has following probability:

$$P(w \mid w_{i-k}, \ldots w_{i-1}) = \frac{1}{|V|}, \qquad \forall w \in V$$

what is the perplexity of the test corpus?

A) $2^{|V|}$ B) $|V|$ C) $|V|^2$ D) $2^{-|V|}$

$$\text{ppl} = 2^{-\frac{1}{n} n \log(1/|V|)} = |V|$$

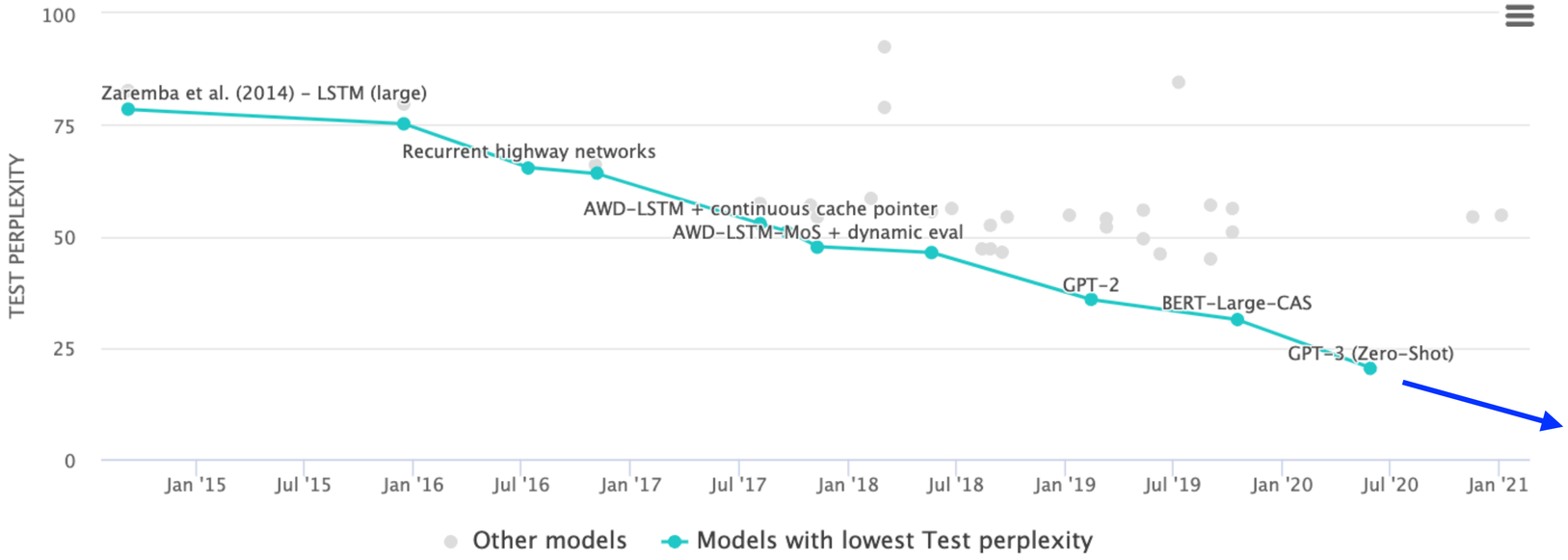
*Measure of model's uncertainty about next word (aka `average branching factor')*

branching factor = # of possible words following any word

# Perplexity

Training corpus 38 million words, test corpus 1.5 million words, both **WSJ**

| N-gram Order | Unigram | Bigram | Trigram |
|---|---|---|---|
| Perplexity (test) | 962 | 170 | 109 |




GPT-3 175B:
ppl = 20.5

https://paperswithcode.com/sota/language-modelling-on-penn-treebank-word

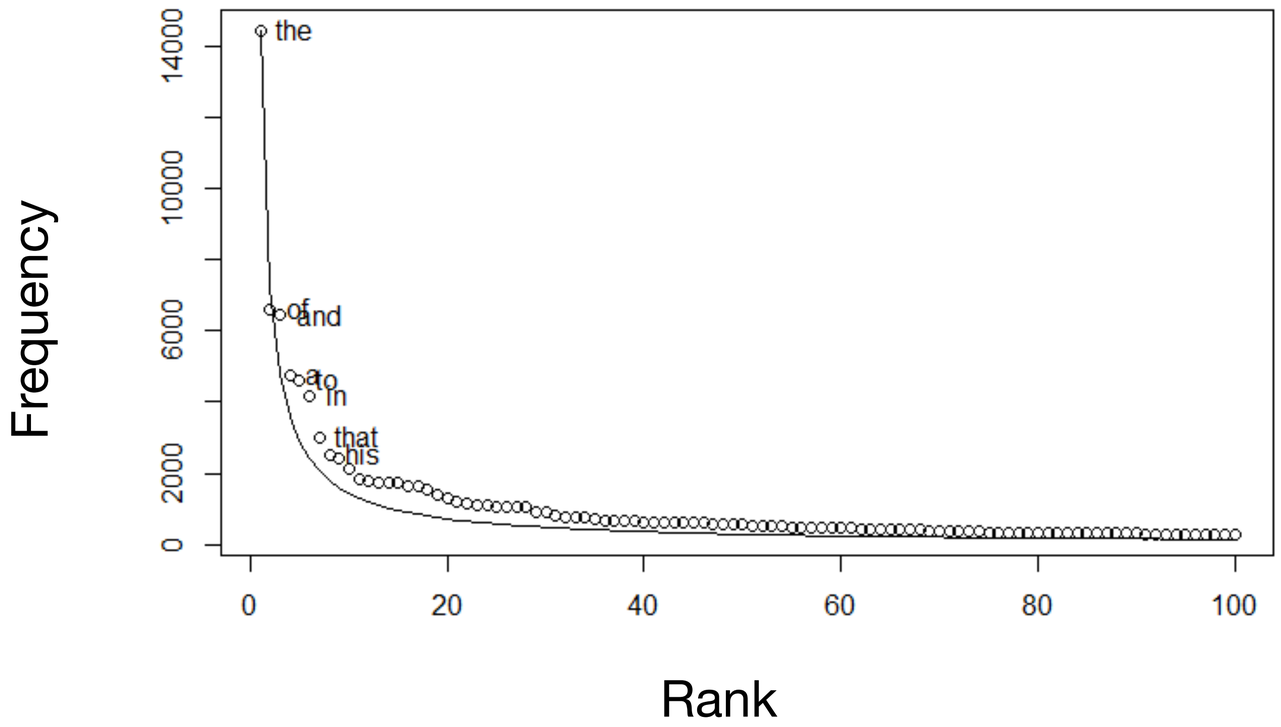
# Smoothing

# Generalization of n-grams

- Not all n-grams in the test set will be observed in training data
- Test corpus might have some that have zero probability under our model
  - **Training set**: *Google news*
  - **Test set**: *Shakespeare*
  - P(affray | voice doth us) = 0 $\implies$ P(test corpus) = 0
  - Perplexity is not defined.

$$\text{ppl}(S) = 2^x \quad \text{where}$$
$$x = -\frac{1}{n}\sum_{i=1}^{n} \log P(w_i \mid w_1 \ldots w_{i-1})$$

# Sparsity in language




$$freq \propto \frac{1}{rank}$$

Zipf's Law

- Long tail of infrequent words
- Most finite-size corpora will have this problem.

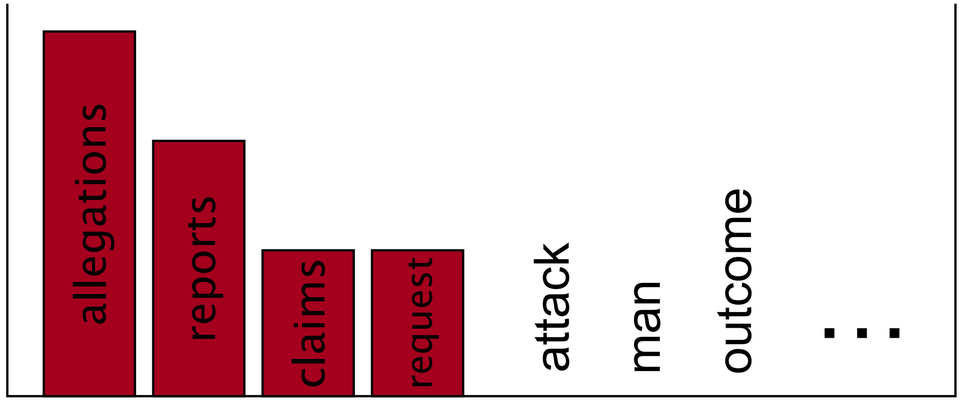
# Smoothing

- Handle sparsity by making sure all probabilities are non-zero in our model
  - **Additive**: Add a small amount to all probabilities
  - **Interpolation**: Use a combination of different granularities of n-grams
  - **Discounting**: Redistribute probability mass from observed n-grams to unobserved ones

# Smoothing intuition

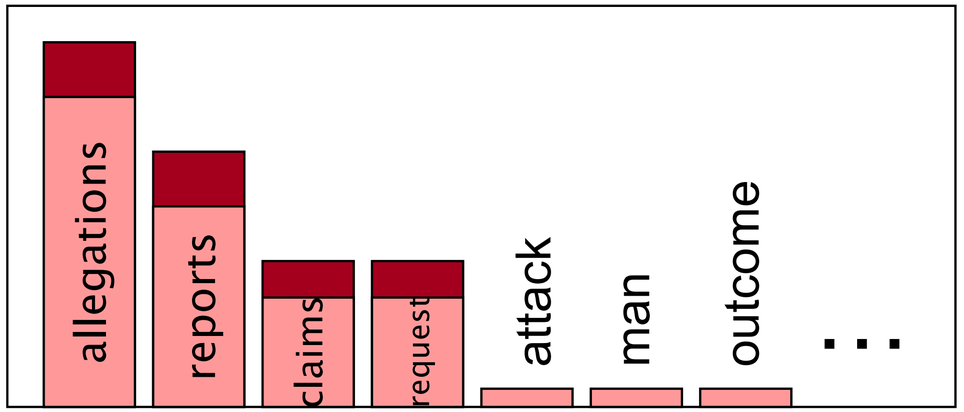
When we have sparse statistics:

P(w | denied the)
3 allegations
2 reports
1 claims
1 request
7 total




Steal probability mass to generalize better

P(w | denied the)
2.5 allegations
1.5 reports
0.5 claims
0.5 request
2 other
7 total




*(Slide credit: Dan Klein)*

# Laplace smoothing

- Also known as add-alpha
- Simplest form of smoothing: Just add $\alpha$ to all counts and renormalize!
- Max likelihood estimate for bigrams:

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i)}{C(w_{i-1})}$$

- After smoothing:

$$P(w_i \mid w_{i-1}) = \frac{C(w_{i-1}, w_i) + \alpha}{C(w_{i-1}) + \alpha |V|}$$

# Raw bigram counts (Berkeley restaurant corpus)

- Out of 9222 sentences

| | i | want | to | eat | chinese | food | lunch | spend |
|---|---|---|---|---|---|---|---|---|
| i | 5 | 827 | 0 | 9 | 0 | 0 | 0 | 2 |
| want | 2 | 0 | 608 | 1 | 6 | 6 | 5 | 1 |
| to | 2 | 0 | 4 | 686 | 2 | 0 | 6 | 211 |
| eat | 0 | 0 | 2 | 0 | 16 | 2 | 42 | 0 |
| chinese | 1 | 0 | 0 | 0 | 0 | 82 | 1 | 0 |
| food | 15 | 0 | 15 | 0 | 1 | 4 | 0 | 0 |
| lunch | 2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| spend | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |

*(Slide credit: Dan Jurafsky)*

# Smoothed bigram counts

| | i | want | to | eat | chinese | food | lunch | spend |
|---|---|---|---|---|---|---|---|---|
| i | 6 | 828 | 1 | 10 | 1 | 1 | 1 | 3 |
| want | 3 | 1 | 609 | 2 | 7 | 7 | 6 | 2 |
| to | 3 | 1 | 5 | 687 | 3 | 1 | 7 | 212 |
| eat | 1 | 1 | 3 | 1 | 17 | 3 | 43 | 1 |
| chinese | 2 | 1 | 1 | 1 | 1 | 83 | 2 | 1 |
| food | 16 | 1 | 16 | 1 | 2 | 5 | 1 | 1 |
| lunch | 3 | 1 | 1 | 1 | 1 | 2 | 1 | 1 |
| spend | 2 | 1 | 2 | 1 | 1 | 1 | 1 | 1 |

Add 1 to all the entries in the matrix

*(Slide credit: Dan Jurafsky)*

# Smoothed bigram probabilities

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}, w_i) + \alpha}{C(w_{i-1}) + \alpha|V|} \qquad \alpha = 1$$

| | i | want | to | eat | chinese | food | lunch | spend |
|---|---|---|---|---|---|---|---|---|
| i | 0.0015 | 0.21 | 0.00025 | 0.0025 | 0.00025 | 0.00025 | 0.00025 | 0.00075 |
| want | 0.0013 | 0.00042 | 0.26 | 0.00084 | 0.0029 | 0.0029 | 0.0025 | 0.00084 |
| to | 0.00078 | 0.00026 | 0.0013 | 0.18 | 0.00078 | 0.00026 | 0.0018 | 0.055 |
| eat | 0.00046 | 0.00046 | 0.0014 | 0.00046 | 0.0078 | 0.0014 | 0.02 | 0.00046 |
| chinese | 0.0012 | 0.00062 | 0.00062 | 0.00062 | 0.00062 | 0.052 | 0.0012 | 0.00062 |
| food | 0.0063 | 0.00039 | 0.0063 | 0.00039 | 0.00079 | 0.002 | 0.00039 | 0.00039 |
| lunch | 0.0017 | 0.00056 | 0.00056 | 0.00056 | 0.00056 | 0.0011 | 0.00056 | 0.00056 |
| spend | 0.0012 | 0.00058 | 0.0012 | 0.00058 | 0.00058 | 0.00058 | 0.00058 | 0.00058 |

*(Credits: Dan Jurafsky)*

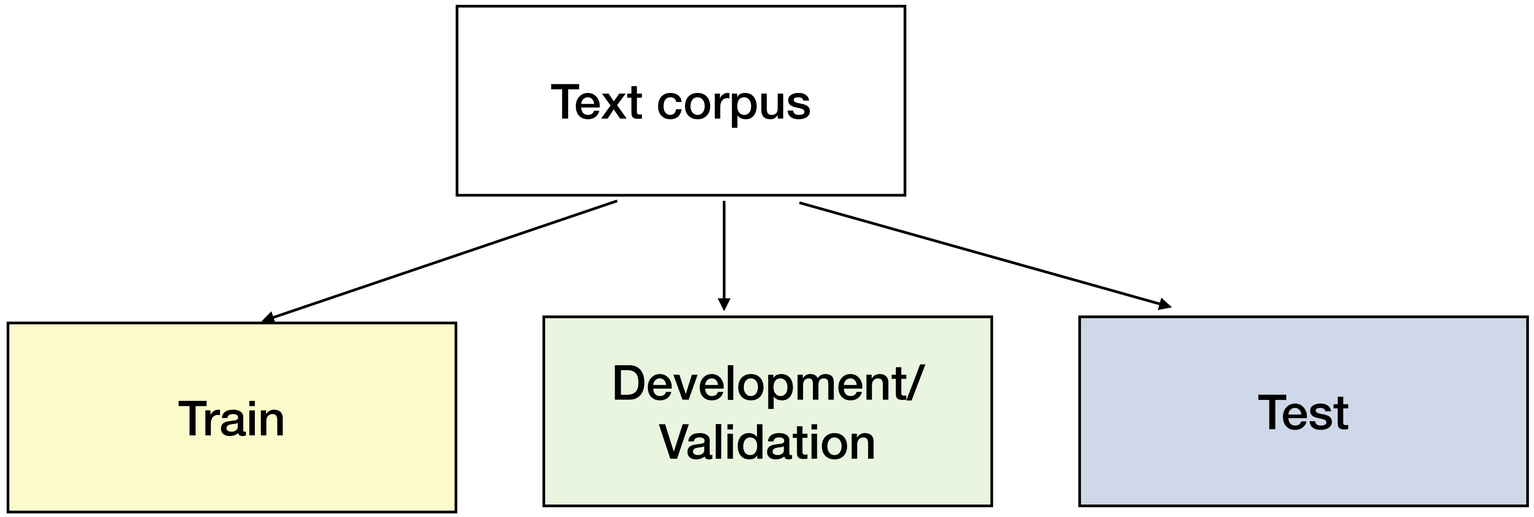
# Linear Interpolation

$$\hat{P}(w_i \mid w_{i-2}, w_{i-1}) = \lambda_1 P(w_i \mid w_{i-2}, w_{i-1}) \quad \text{Trigram}$$
$$+\lambda_2 P(w_i | w_{i-1}) \quad \text{Bigram}$$
$$+\lambda_3 P(w_i) \quad \text{Unigram}$$

$$\sum_i \lambda_i = 1$$

- Use a combination of models to estimate probability
- Strong empirical performance

# How can we choose lambdas?




- First, estimate n-gram prob. on training set
- Then, estimate lambdas (hyperparameters) to maximize probability on the held-out development/validation set
- Use best model from above to evaluate on test set

# Up next: Text classification