Washington University in St. Louis
James McKelvey School of Engineering

CSE 561A: Large Language Models

Fall 2024

Lecture 1: Course Overview

Jiaxin Huang

Content

- **Course Logistics**
- Language Model Basics
- Covered Topics Preview

Course Logistics

- Instructor: Jiaxin Huang (jiaxinh@wustl.edu)
- Teaching Assistant:
  - Chengsong Huang (chengsong@wustl.edu)

- Course meeting times: 2:30pm – 3:50pm Tuesday / Thursday
- Location: Crow / 206

Course Logistics

- Course Syllabus: https://teapot123.github.io/CSE561A_2024fl/
- Canvas: https://wustl.instructure.com/courses/133999 (will be published soon)
- We will be using Canvas for announcements, and project report submissions, and Piazza for discussions.

Course Structure

- Advanced Research-Oriented Course
  - Pre-requisites: Students are expected to understand concepts in machine learning (CSE 417T/517A)
  - We will be teaching and discussing state-of-the-art papers about large language models
  - Lectures of fundamentals of Large Language Models (language model architecture and training framework)
  - Lectures of Large Language Model Capabilities, Applications and Issues
    - This part consists of a list of frontier research papers (will be released later), from which students will choose their interested papers to present in the class
    - Students who are not presenters are expected to participate in discussion and submit 4 preview questions
  - Guest lectures on frontier research topics

Grading

- 15% Class Participation
  - Regular class participation and discussion (10%)
  - Preview question submissions (5%)
- 30% Class Presentation
- 55% Final Project
  - 10% Project Proposal
  - 10% Mid-term Report
  - 10% Final Course Presentation (Group-Based)
  - 5% Feedbacks for other groups’ final project presentations
  - 20% Final Project Report

In Class Presentation

- Starting from Week 3, each lecture will consist of one research topic of large language models, with 4 state-of-the-art papers. Each lecture will be covered by two students.
- Each student is required to do a 30-min presentation in class to cover two papers, followed by a 5-min Q&A/discussion session.
- Sign-up sheet for paper presentation will be released later this week.
- Remember to send over your slides to the instructor (and cc the TA) before your presentation:
  - For Tuesday classes, send over your slides before the previous Friday 12:00PM
  - For Thursday classes, send over your slides before the previous Monday 12:00PM
- When it is not your turn to present, you can preview the paper in advance. Each student is required to submit 4 preview questions for **4** times (need to be on **4 different dates**). Each preview question is submitted for a paper one day before the presentation. You are also encouraged to raise that question in class.
  - Preview questions cannot be simple ones like “what is the aim of the paper?”

In Class Presentation

- How to present a paper?
  - Think about the context of the research: introduce the background of the research topic
  - What is the challenge and contribution of this paper, given the research background?
  - The method: from framework to technical details
  - What are some interesting experiment results and observations?
  - What could be done in the future?
  - Summarize the takeaways/highlights of this paper

In Class Presentation

- More tips to do presentations
  - Get familiar with your material. Don't read scripts for the whole time.
  - Make eye contact with audiences.
  - Make your voice loud enough so that everyone can hear you clearly
  - Please control your time(30min)! We will give you notice when your time is nearly used up.

Final Project

- Students need to form groups of 2-3 people to do a large language model research project.
- Project proposal deadline: 9/16 11:59PM
- Midterm project report deadline: 10/21 11:59PM
- Final project presentation deadline: 12/2 11:59PM
  - We will use two lectures for project presentation: 12/3, 12/5
- Final project report deadline: 12/13 11:59PM

Final Project

- There are typically two types of projects.
- 1) Designing a novel algorithm to train a medium-sized language model: BERT, GPT-2 for problems that you are interested in.
  - https://huggingface.co/models
- 2) Designing a novel algorithm to do inference on large language models (white box models such as LLaMA2 models, or black box models such as GPT-4, CLAUDE, etc.) to solve some type of complex problems, and analyze their limitations. (We may not be able to reimburse for the API costs)
  - https://platform.openai.com/docs/introduction
  - https://docs.anthropic.com/claude/reference/getting-started-with-the-api

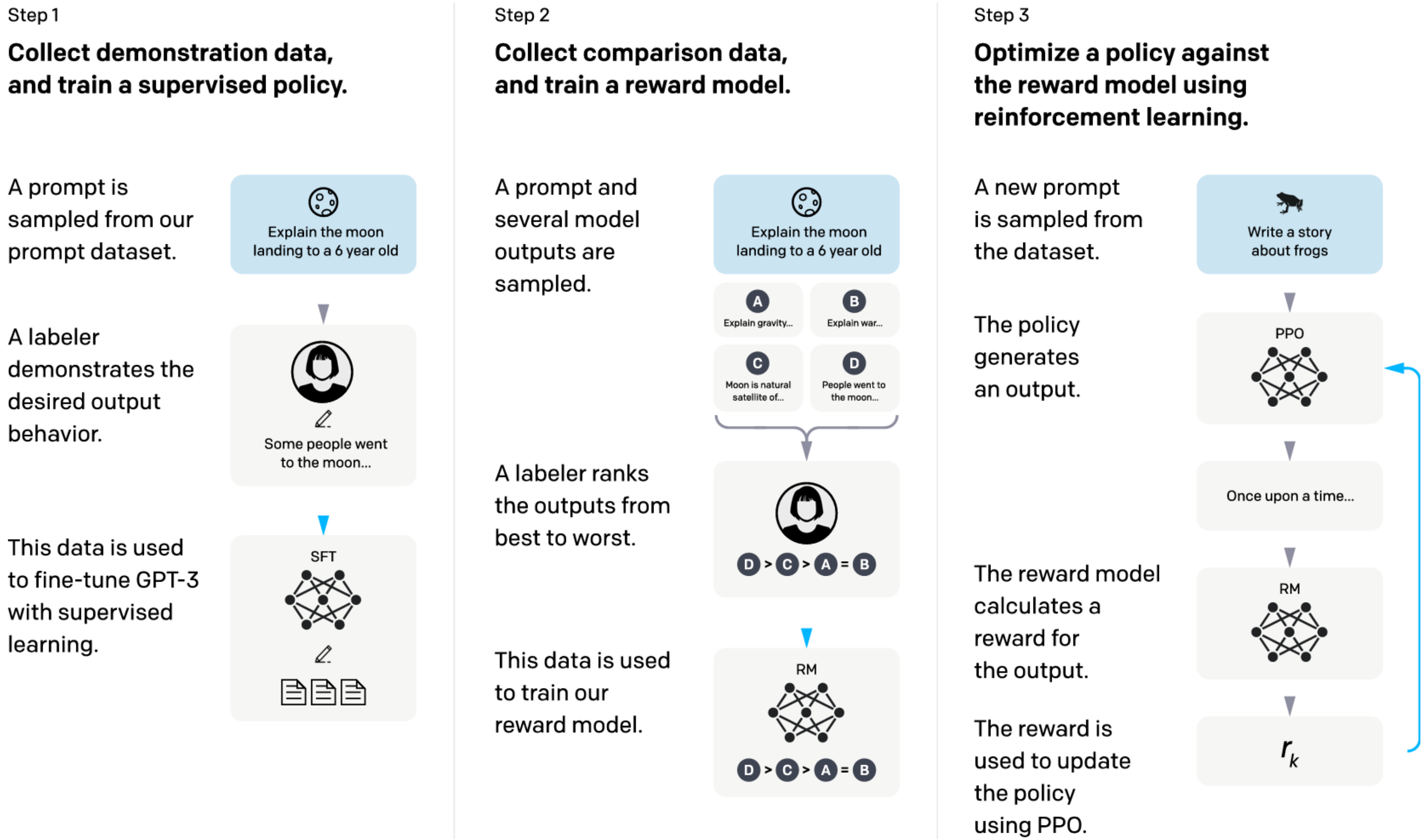
Final Project Presentation

- Near the end of the semester, we will create a signup sheet for the final project presentation.
- We anticipate to distribute project presentations into two lectures (12/3, 12/5), and you will need to signup for a time slot.
- Length of project presentation: 5-8min depending on the number of groups
- Students will need to submit feedback scores for other groups' presentation (through Google Form).

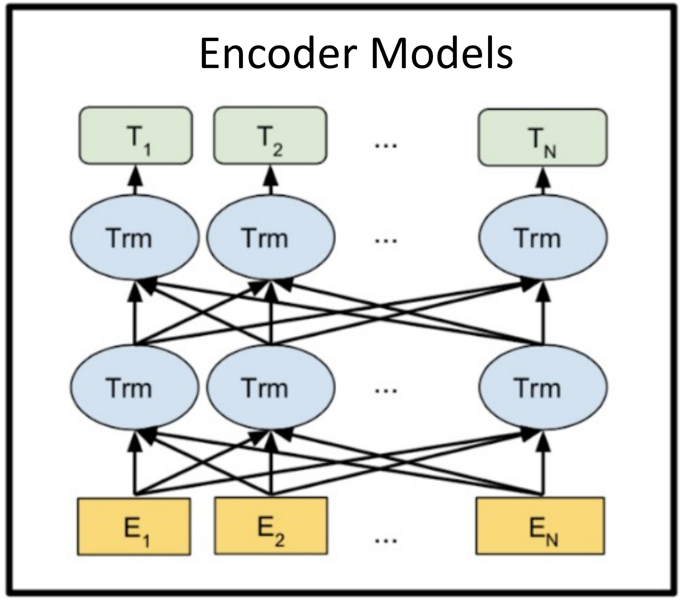
Content

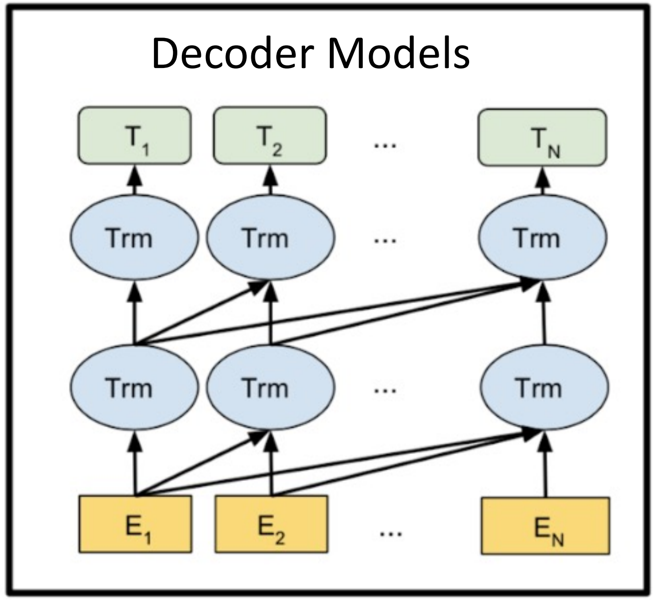
- Course Logistics
- **Covered Topics Preview**
- Language Model Basics

Large Language Model Pre-training Framework

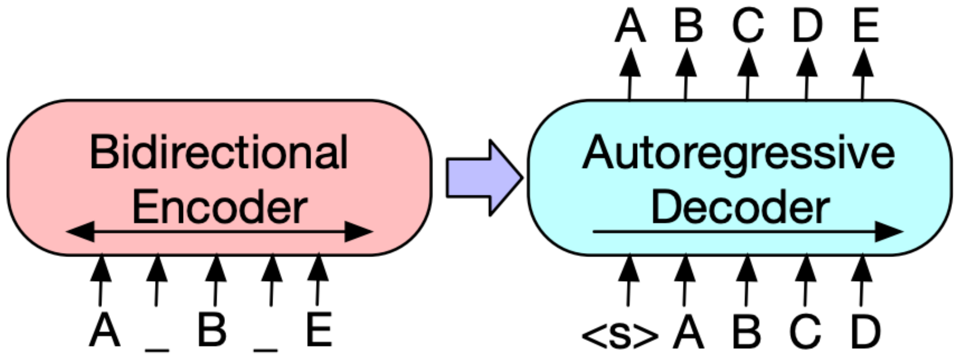


Language Model Architectures (will be covered in the next course)





Encoder-Decoder Models



Topics: Language Model Reasoning

- Google's Minerva Model

Minerva

Question: Assume that the variance of the first n natural numbers is 10, and the variance of the first m even natural numbers is 16. Compute $m+n$.

Model output: Let σ^2 be the variance of the first m even natural numbers, and τ^2 be the variance of the first n natural numbers. Then, we have $\tau^2 = \frac{n^2-1}{12}$ and $\sigma^2 = \frac{m^2-1}{3}$. We solve the system of equations:

$$\frac{n^2-1}{12} = 10$$
$$\frac{m^2-1}{3} = 16.$$

This translates into the system

$$n^2 = 121$$
$$m^2 = 48+1 = 49.$$

Therefore, $n = \boxed{11}$ and $m = \boxed{7}$, so $n+m = \boxed{18}$.

Question: For every $a, b, b \neq a$ prove that

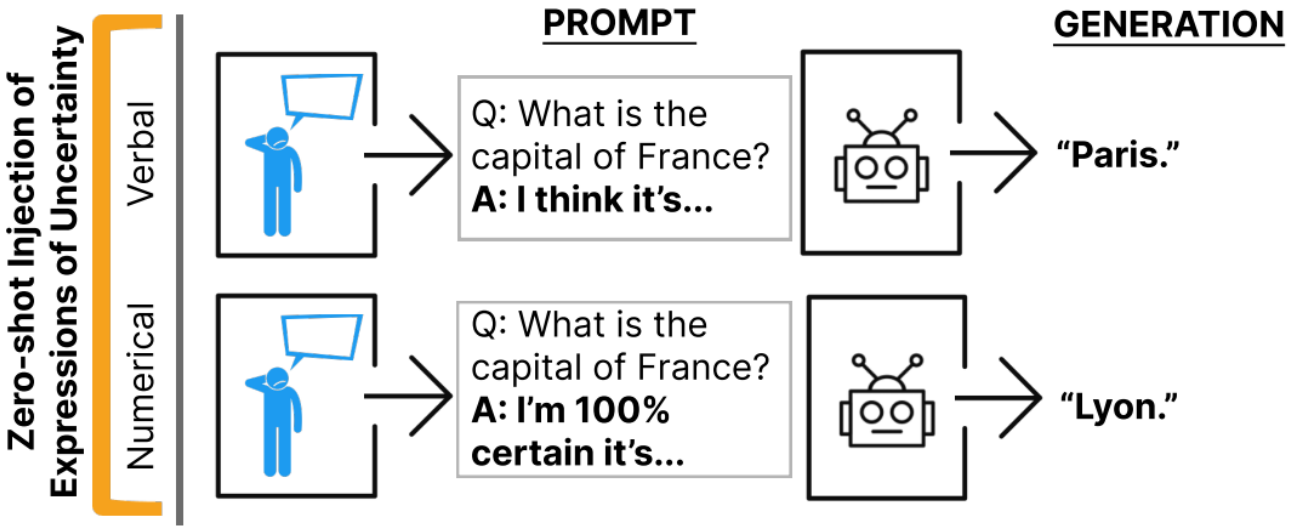
$$\frac{a^2+b^2}{2} > \left(\frac{a+b}{2}\right)^2.$$

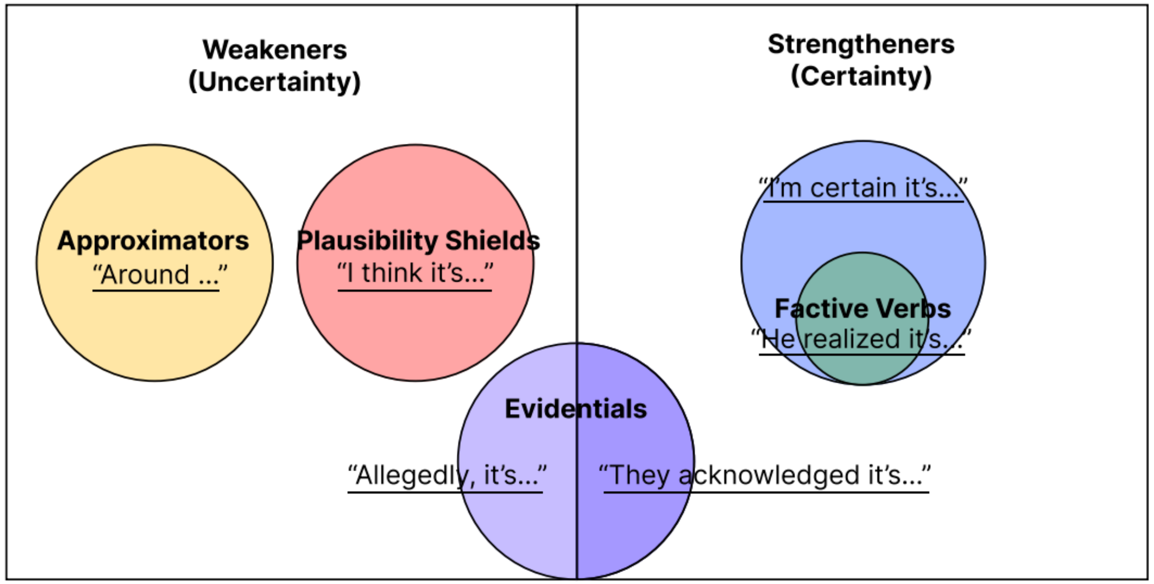
Model output:

$$\frac{a^2+b^2}{2} > \left(\frac{a+b}{2}\right)^2$$
$$\iff \frac{a^2+b^2}{2} > \frac{a^2+b^2+2ab}{4}$$
$$\iff a^2+b^2 > \frac{a^2+b^2+2ab}{2}$$
$$\iff 2a^2+2b^2 > a^2+b^2+2ab$$
$$\iff a^2+b^2 > 2ab$$
$$\iff a^2+b^2-2ab > 0$$
$$\iff (a-b)^2 > 0$$

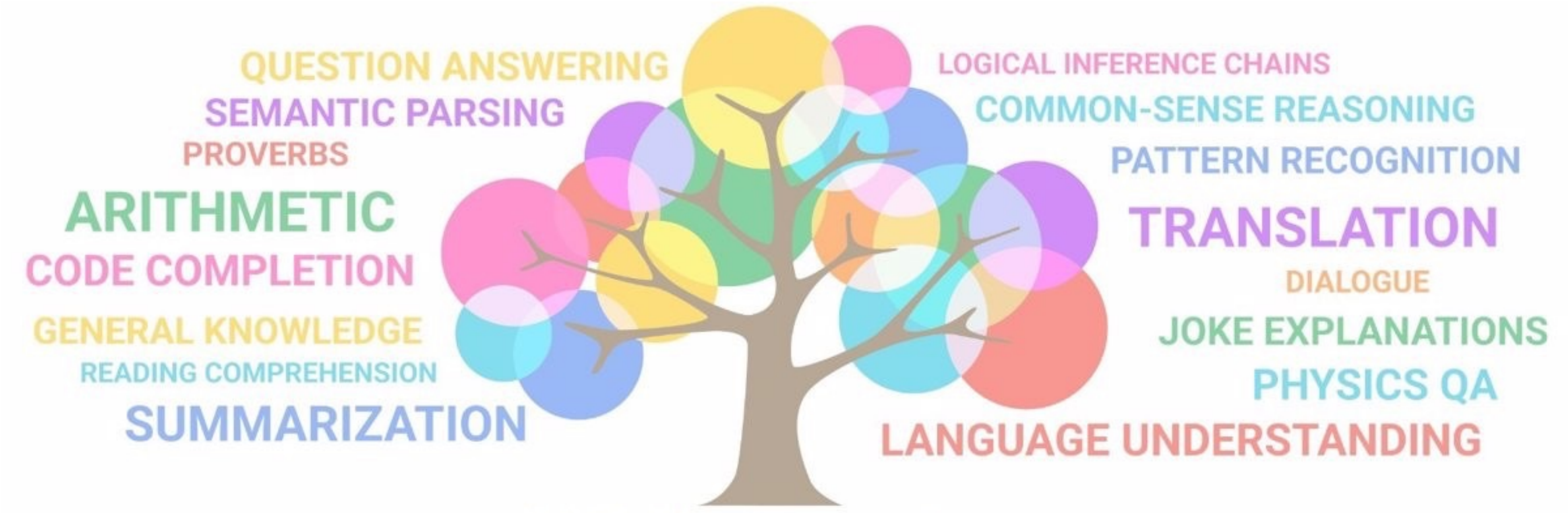
which is true, because the square of a real number is positive.

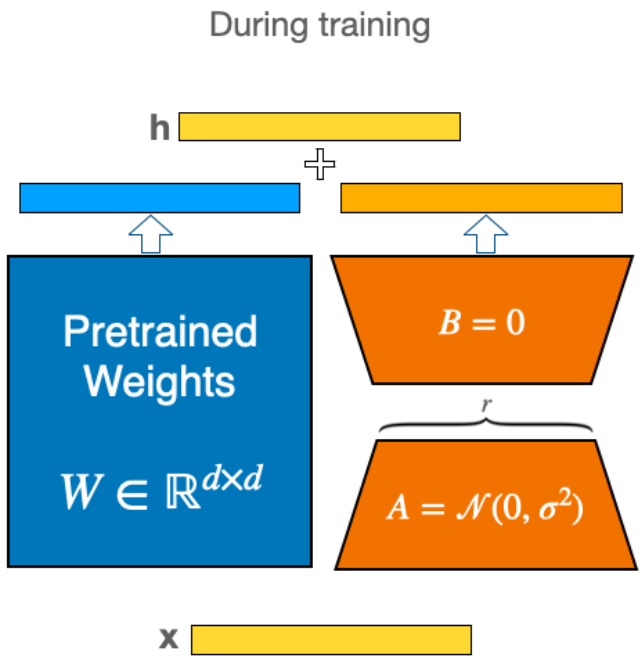
Topics: Language Model Calibration





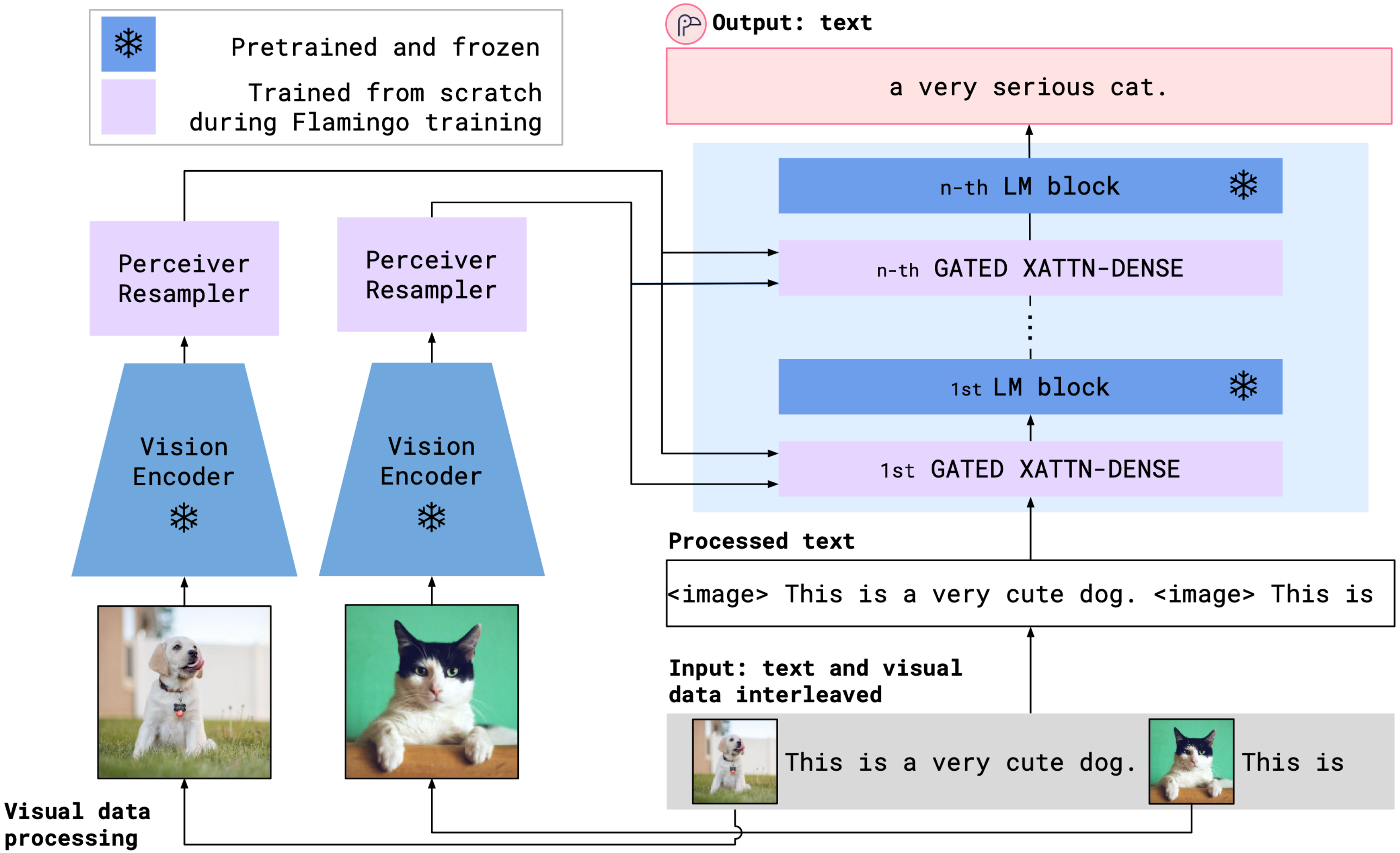
Topics: Efficient Fine-Tuning



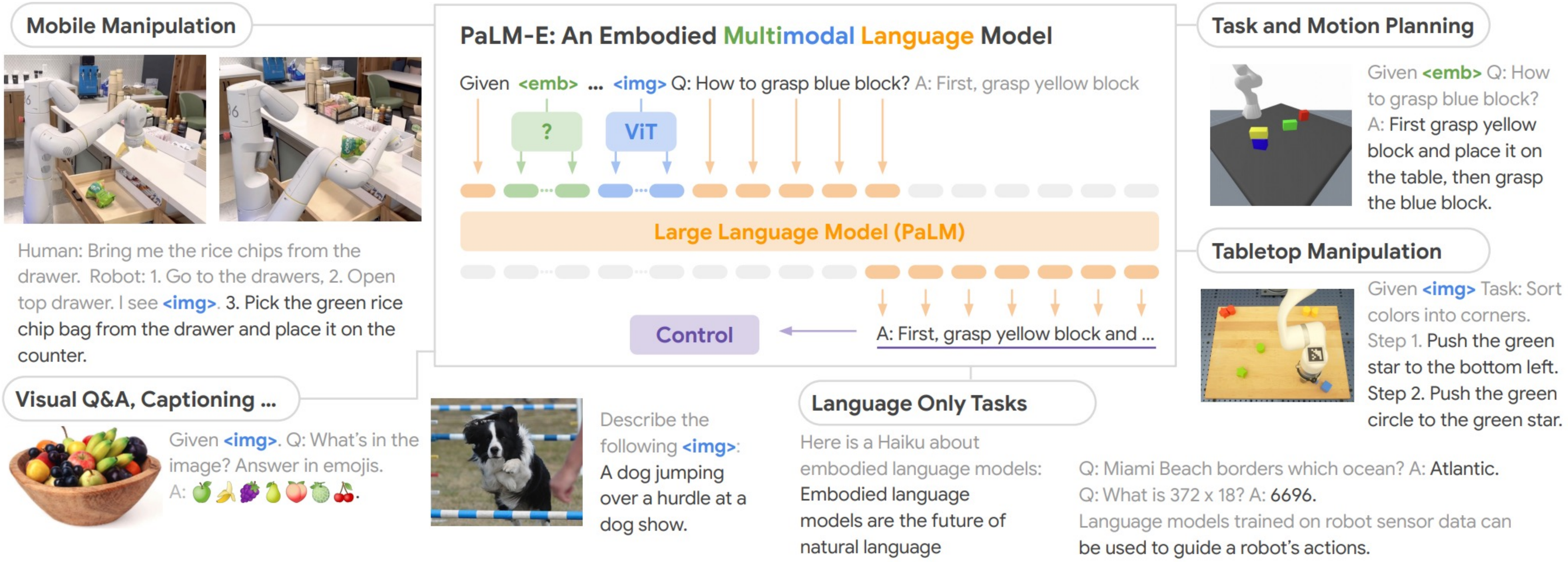


Unsupervised/Self-supervised;
On large-scale general domain corpus → Task-specific supervision;
On target corpus

Topics: Multimodal Language Model

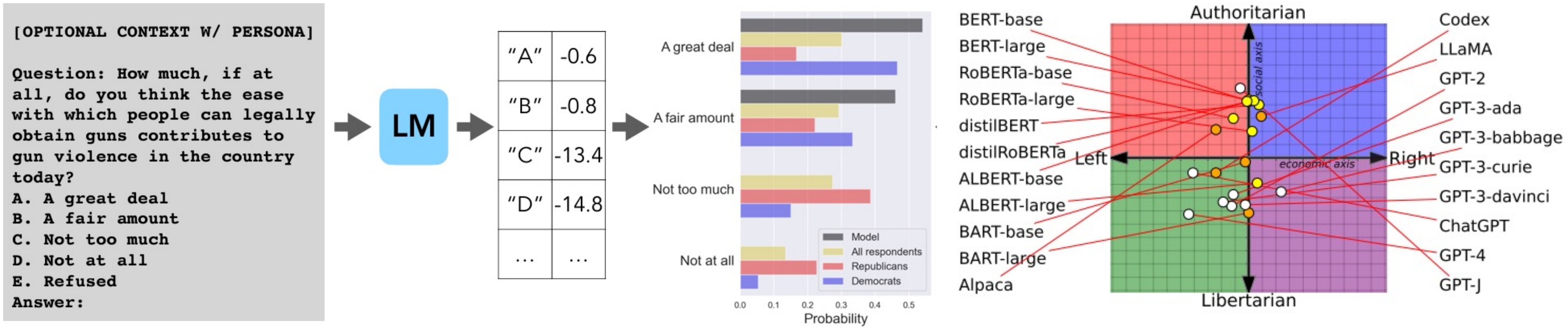


Topics: Language Model as Agents



Topics: Bias of Language Models

- Different language models may have different political views.



Content

- Course Logistics
- Covered Topics Preview
- **Language Model Basics**

What are language models?

Language models

- The classic definition of a language model (LM) is a probability distribution over each token sequence $[w_1, w_2, \ldots, w_n]$, whether it's a good or bad one.
- Sally fed my cat with meat: P(Sally, fed, my, cat, with, meat) = 0.03,
- My cat fed Sally with meat: P(My, cat, fed, Sally, with, meat) = 0.005,
- fed cat Sally meat my with: P(fed, cat, Sally, meat, my, with) = 0.0001

Autoregressive language models

- The chain rule of probability:
- P(Sally, fed, my, cat, with, meat) = P(Sally)

  * P(fed | Sally)

  * P(my | Sally, fed)

  * P(cat | Sally, fed, my)

  * P(with | Sally, fed, my, cat)

  * P(meat | Sally, fed, my, cat, with)

Conditional probability

$$p(w_1, w_2, w_3, \ldots, w_N) =$$
$$p(w_1)\; p(w_2|w_1)\; p(w_3|w_1, w_2) \times \ldots \times p(w_N|w_1, w_2, \ldots w_{N-1})$$

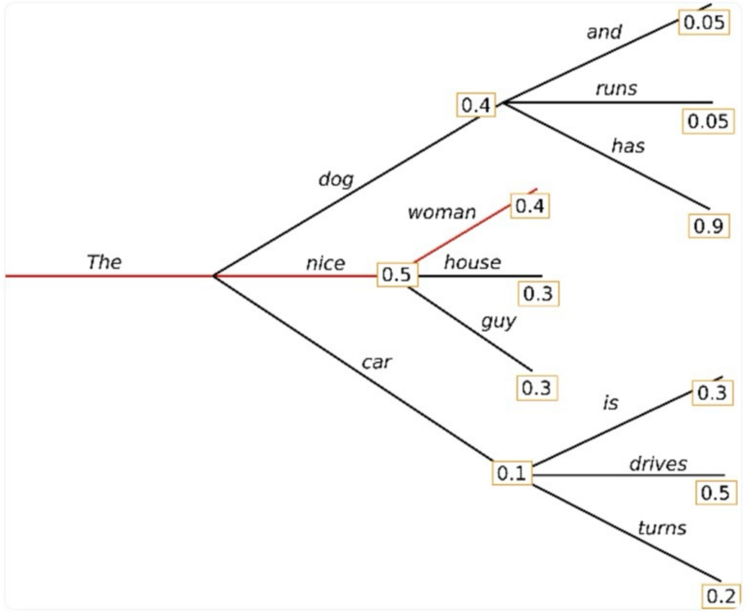
Sequence generation with language model

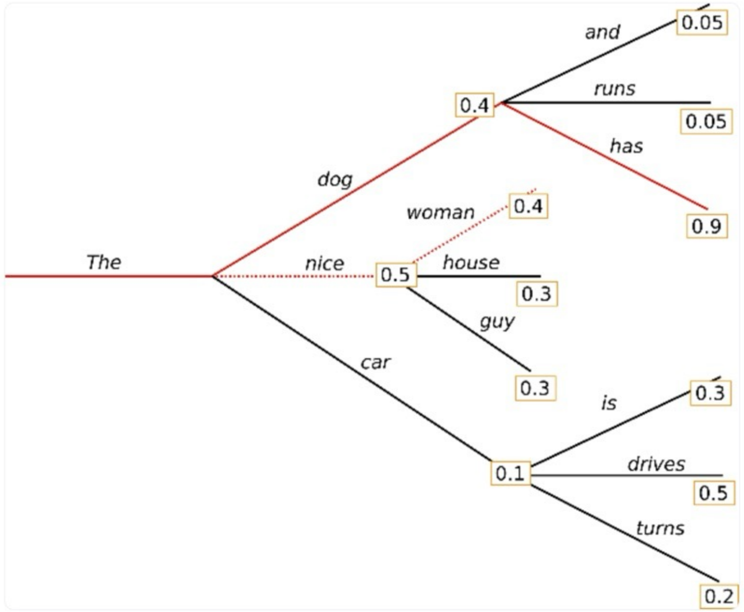
- If we already have a good language model, a given text prompt $w_{[1:n]}$, and we want the model to generate a good sentence completion with the length of L: How to find $w_{[n+1:n+L]}$ with the highest probability?
- Enumerate over all possible combinations?

- Next token prediction: generating the next token step by step, starting from w_{n+1} using $p\left(w_{n+1} \middle| w_{[1:n]}\right)$
- To select the next token with $p\left(w_{n+1} \middle| w_{[1:n]}\right)$, there are also different decoding approaches.

Different Decoding Approaches

- Greedy decoding: At each step, always select w_t with the highest $p(w_t|w_{[1:t-1]})$.
- Beam Search: Keep track of k possible paths at each step instead of just one. Reasonable beam size k: 5-10 .



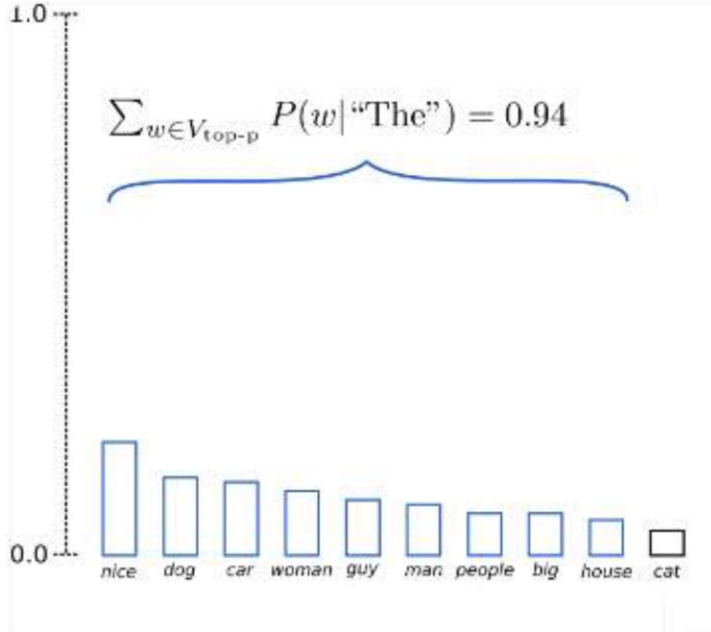


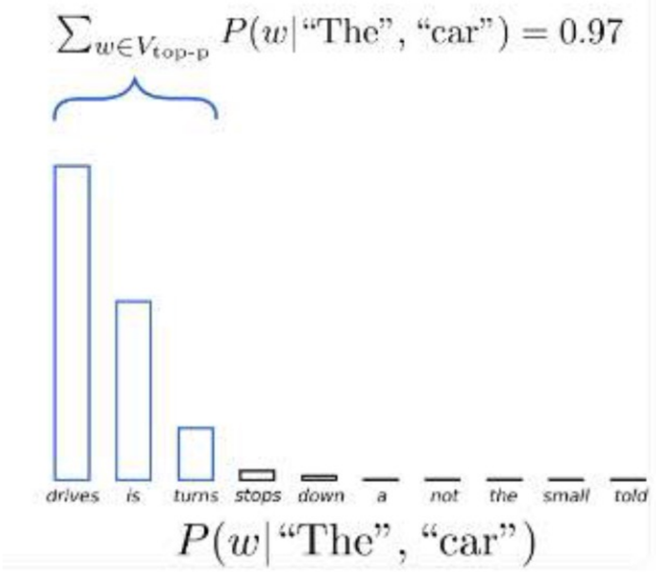
Different Decoding Approaches

- Top-k sampling: At each step, randomly sample the next token from $p\left(w_t \middle| w_{[1:t-1]}\right)$, but restrict to only the k most probable tokens.
- Allows you to control diversity:
  - Increase k gives you more creative / risky outputs.
  - Decrease k gives you safer outputs.

- Top-p sampling: At each step, randomly sample the next token from $p\left(w_t \middle| w_{[1:t-1]}\right)$, but restrict to the set of tokens with a cumulative probability of p
  - throw away long-tailed tokens

- Top-k and Top-p can be used together!





Q: How to train a good language model?

Q: How to train a good language model?

A: Maximizing the language model probability of an observed large corpus.

N-gram Language Models

- Bigram models

	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

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

$P(\texttt{i}|\texttt{<s>}) = 0.25$ $\quad$ $P(\texttt{english}|\texttt{want}) = 0.0011$

$P(\texttt{food}|\texttt{english}) = 0.5$ $\quad$ $P(\texttt{</s>}|\texttt{food}) = 0.68$

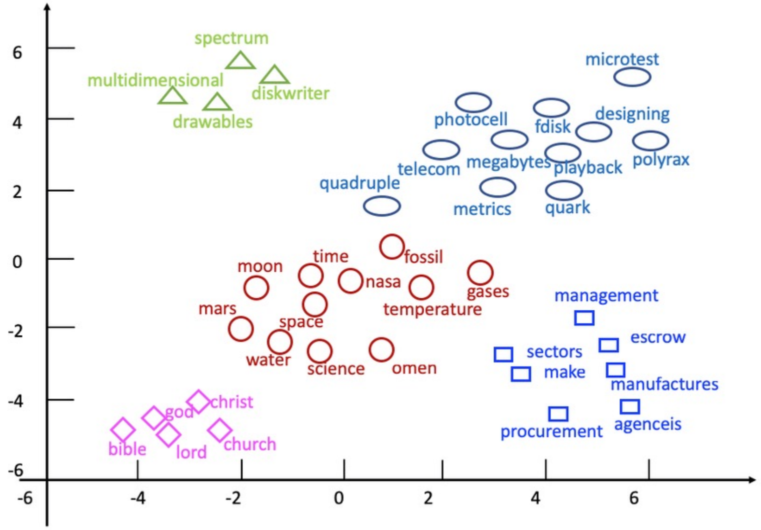
<s> is the starting token of a sentence.
</s> is the ending token of a sentence.

Curse of Dimensionality

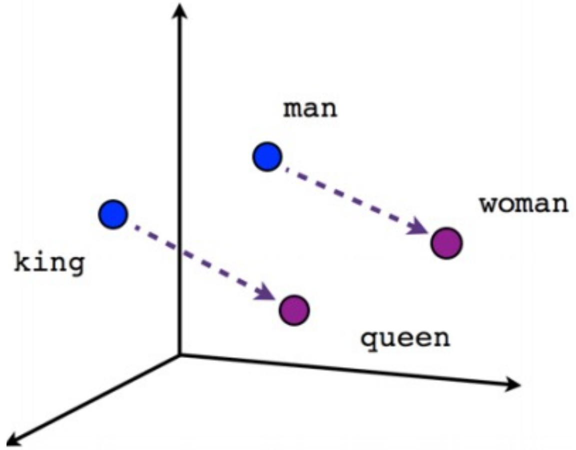
- Limitation of N-gram models
  - Limited Context Length: N-grams have a finite context window of length N, which means they cannot capture long-range dependencies or context beyond the previous N-1 words
  - Sparsity: As N increases, the number of possible N-grams grows exponentially, leading to sparse data and increased computational demands
    - Suppose vocabulary size is V, the number of possible N-grams increases to V^N.
  - Usually V (vocabulary size) could be more than ten thousand. Representing each word as a one-hot vector is inefficient.
    - “Dogs” and “cats” are more similar, compared to “dogs” and “rectangular”.

How to represent text more efficiently?

- Word Embedding: A milestone in NLP and ML
  - Unsupervised learning of text representations—No supervision needed
  - Embed one-hot vectors into lower-dimensional space—Address “curse of dimensionality”
  - Word embedding captures useful properties of word semantics
  - Word similarity: Words with similar meanings are embedded closer
  - Word analogy: Linear relationships between words (e.g. king - queen = man - woman)



Word Similarity



Word Analogy

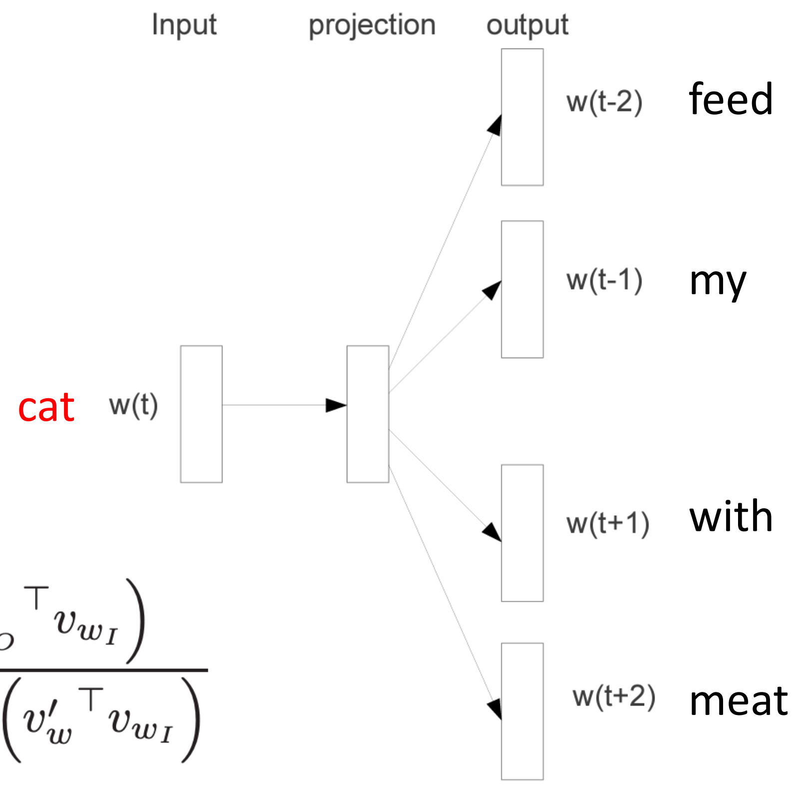
Distributed Representations: Word2Vec

- Assumption: If two words have similar contexts, then they have similar semantic meanings!
- Word2Vec Training objective:
- To learn word vector representations that are good at predicting the nearby words.

Co-occurred words in a **local context window**

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(\boxed{w_{t+j}|w_t})$$

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^\top v_{w_I}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^\top v_{w_I}\right)}$$



Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS.

Considering subwords: fastText

- fastText improves upon Word2Vec by incorporating subword information into word embedding

Tri-gram extraction

<where> → <wh, whe, her, ere, re>

- fastText allows sharing subword representations across words, since words are represented by the aggregation of their n-grams

Word2Vec probability expression

$$p(w_O|w_I) = \frac{\exp\left(\boxed{{v'_{w_O}}^\top v_{w_I}}\right)}{\sum_{w=1}^{W}\exp\left({v'_w}^\top v_{w_I}\right)} \longrightarrow \sum_{g\in\mathcal{G}_w} \boxed{\mathbf{z}_g^\top}\,\mathbf{v}_c.$$

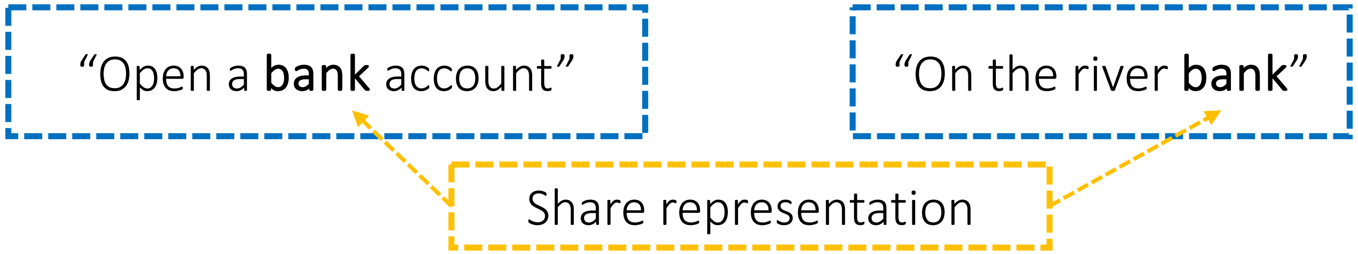
Represent a word by the sum of the vector representations of its n-grams

N-gram embedding

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. Transactions of the Association for Computational Linguistics, 5, 135-146.

Limitations of Word2Vec embeddings

- 1) They are **context-free** embeddings: each word is mapped to only one vector regardless of its context!
    - E.g. “bank” is a polysemy, but only has one representation



- 2) It does not consider the order of words
- 3) It treats the words in the context window equally

Next Lecture: Self-Attention and Transformers