- Review of LLM Talk by Vishal Misra
- Cofounded Cricinfo
- ESPN acquired Cricinfo later, and kept the original interface
- Used GPT-3 to create text2sql interface. AskCricInfo running in production.
- Take query -> Get intent -> Convert to DSL -> Send to LLM -> Get the answer.
- Based primarily on "in-context learning".
  - No need to train model.
- **How does it work? Why does it work?**

Roadmap

- Focus on training objective of LLMs
- Interpret the text gen process as approximating a (very) large matrix of multinomial distributions.
- Prove a universal representation theorem of multinomial distributions as a linear combination of dirichlet distributions.
- Show the emergence of in-context-learning to be consistent with with Bayesian learning where
  - Prior -> Pre-trained model multinomial distribution
  - Prompt -> new evidence / likelihood
  - Bayesian posterior -> multinomial distribution used in text generation.

ChatGPT

- Ability to perform new tasks from only instructions
- Intuitive chat interface
- Free and open

Training Objective - Language Modelling

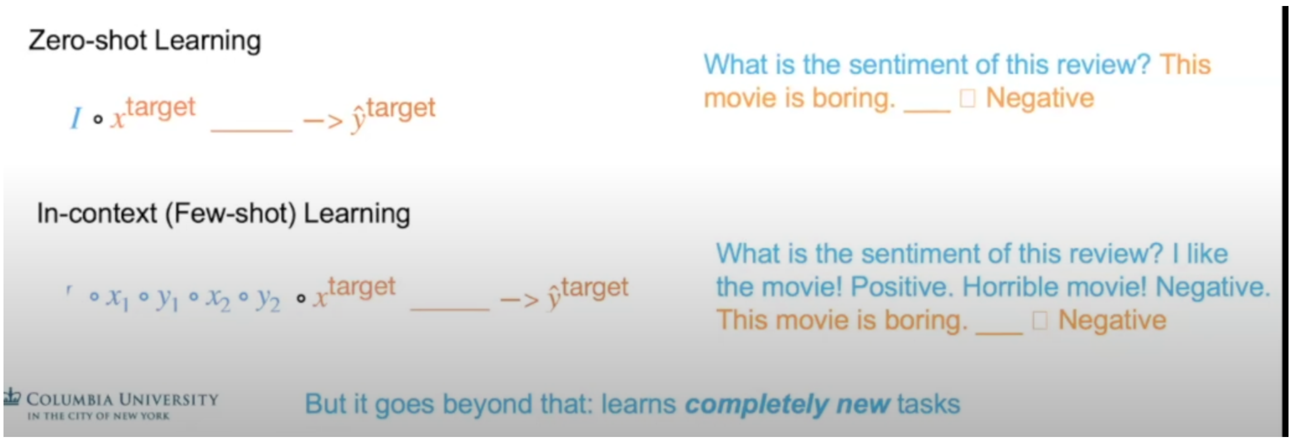
- Predict the next word in a sequence
- Model has vocab. Model produces distribution over words in the vocab.
- Once generated, sample from the distribution.
- Append to the text.
- Repeat

Through this, it learns many concepts such as:

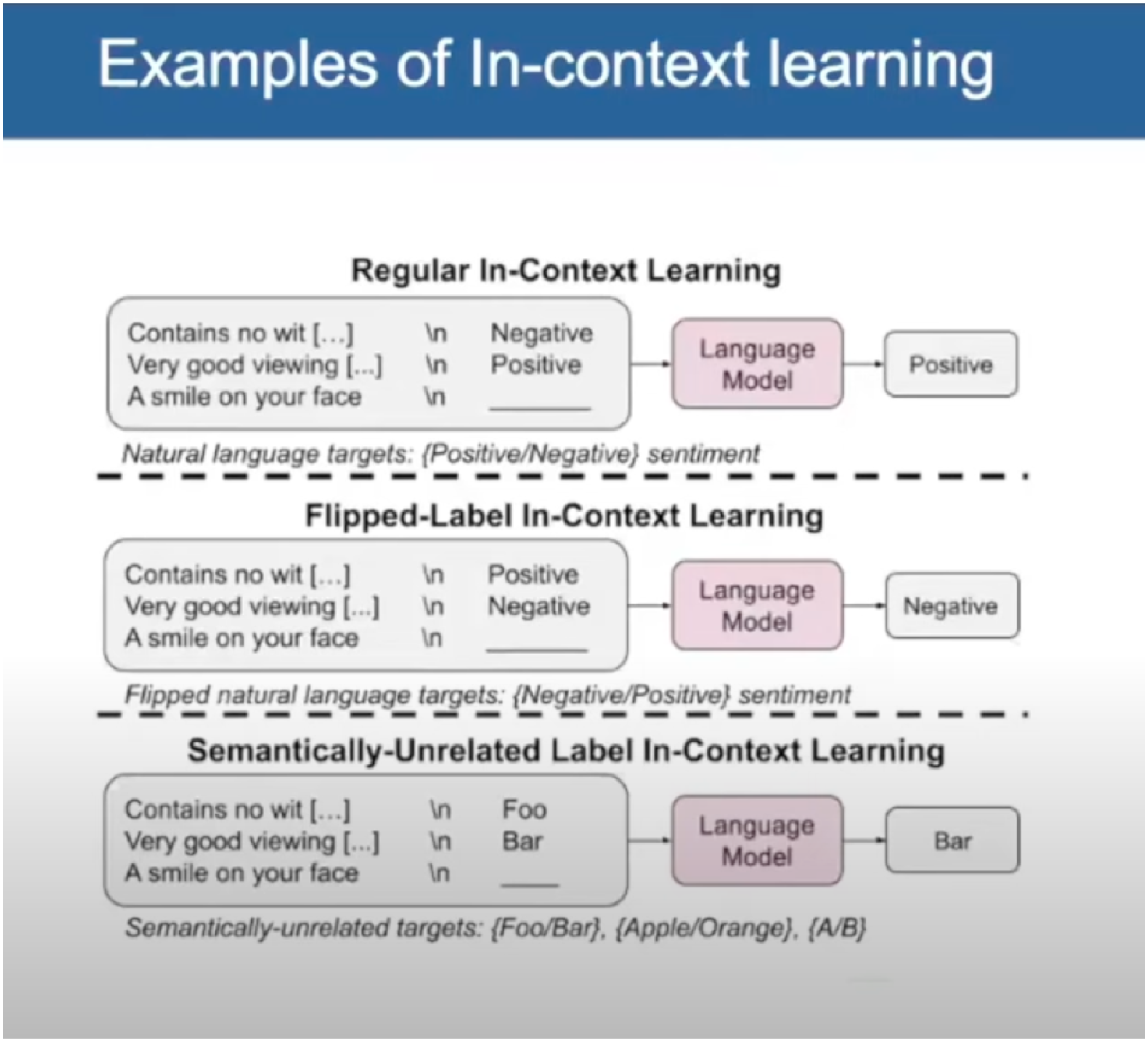
- Grammar
- World Knowledge
- Arithmetic P(2+2 = 4) > P(2+2 = 5)

Test-time => Zero-shot / in-context learning

- Quickly learn new task with no/few labelled examples **without updating model paramteers**
- **Why do we care?**
  - save annotation efforts
  - Change time scale of learning to real time
  - This is the **one true emergent ability** of LLMs



Examples of In-context Learning



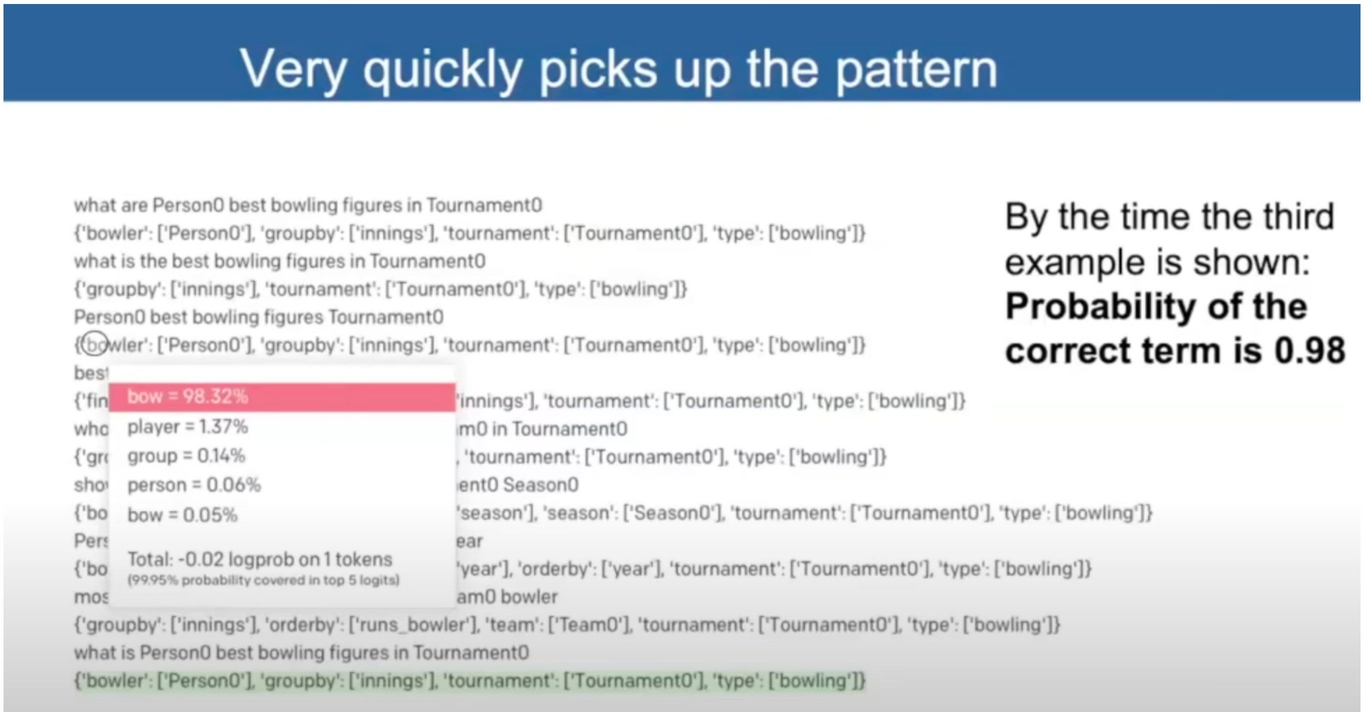
- semantically-unrelated label ICL is the most difficult task, and emerged with LLMs.

Walking through AskCricInfo

- Created MetaLanguage to interpret query intent
  - Query: What are the best bowling figures in an IPL final?
  - Adjusted query: What are the best bownling figures in an Tournament0 final
  - Metalanguage: `{'final_type': ['tournament final'], 'groupby': ['innings'], 'tournament': ['Tournament0'], 'type': ['bowling']}`
  - `Tournament0` used to make it a generic query that can work for any tournament. Replace it before answering with the right tournament name.

"Zero-shot" query to ChatGPT

- What is mohammed siraj's best bowling figures in ODIs?
    - Metalanguage format: Person0's best bowling figures in Tournament0 were 5-20.
- Numbers are completely made up.
- After giving few-shot examples, metalanguage query generated is correct.



LLM Primer

- four kinds of parameter
    - Token size ("vocabulary" of the LLM)
    - Context size ("memory" of the LLM)
    - Parameter count (roughly weights of neural net)
    - Embedding vector (a vector space to represent words/tokens)
- For ChatGPT
    - Token size: ~50000
    - Context size: 8192 tokens for GPT 3.5
    - Parameter count: 175 billion (known for ChatGPT)
    - Embedding vector size: 12880 (recently another version has 1536)

First Generative text model

- Trained by Claude Shannon
- Model based on simple 1st order markov chain
- **LLMs are n'th order Markov Chains, where "n" is the prompt or context length**

Huge probability matrix

- $$\text{Probability matrix size: } 50000^{8000} \ X \ 50000$$
- Each row represents a unique combination of upto 8000 words, from a vocab of 50,000 words
- The column values in each row represent the multinomial distribution to the next word
- The number of rows in this matrix exceeds the number of atoms across across all galaxies....

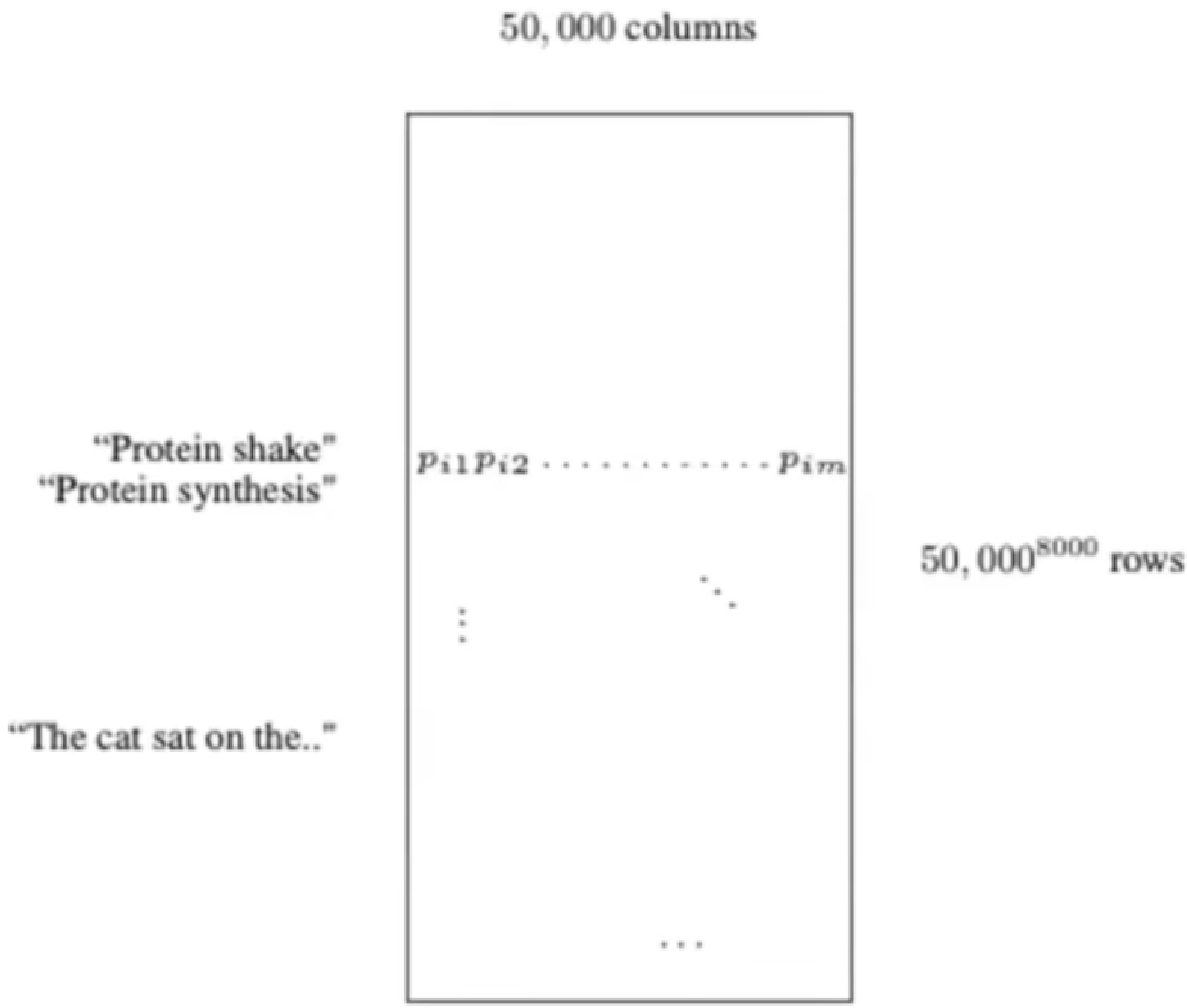
Fortunately, the matrix is extremely sparse

- Most rows individually occur with 0 probability
- Even for rows that occur with relatively high prob in real word, the multinomial distribution row is sparse i.e not all column values will be the same. Eg: "The cat sat on a " is unlikely to be followed by "mRNA"
- Still, 175 billion or even a trillion parameters are not enough to "represent" this matrix
- Use of embeddings further compresses representation

So what are LLMs trying to do?

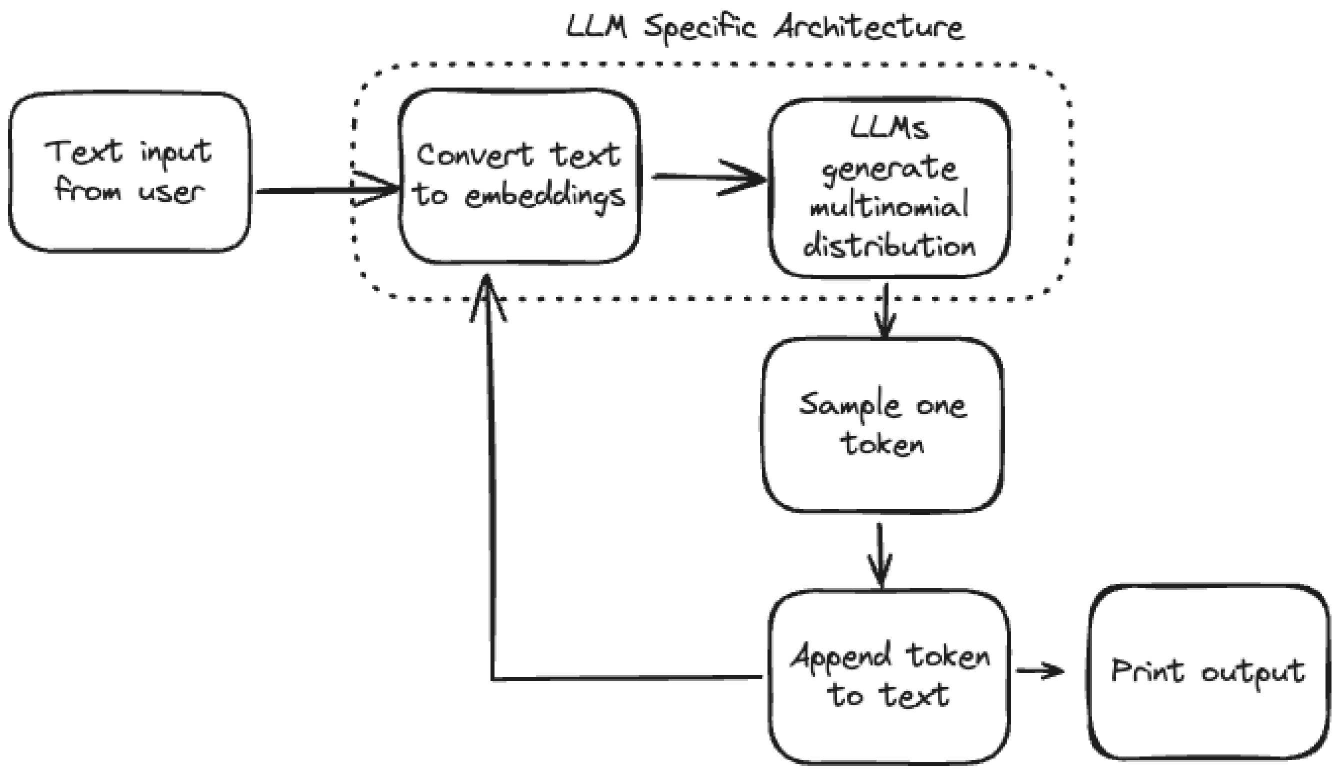
- They are trying to come up with the above matrix representation

The Matrix



Training and Generation of LLMs

- Training process consists of LLMs minimizing the multinomial distribution error of each row P("The cat sat on the mat") based on training data
- In the limit, the generation process reproduces the empirical distribution induced by the training set



Continuity Theorm

Suppose T is a mapping frmo an embedding space to the space of multinomial distributions, and is convexity preserving.

$$T(\alpha e2 + (1-\alpha)e2 = \alpha T(e1) + (1-\alpha)Te2)$$

ELI5: Allows mapping of multinomial distribution from unseen embeddings as a linear combination of mappings of "closest" known embeddings.

Universal representation theorem

Any continuous multinomial distribution

$$u(p_1, p_2, ..p_n)$$

can be approximated as a mixture of dirichlet distributions

$$D(p|k_1 + 1, k_2 + 1..k_m + 1)$$

where

$$\sum k_i = n$$

, each distribution has parameters **k**

$$p(\theta|k) = \frac{1}{B(k)} \prod_{i=1}^{m} \theta_i^{k-1}$$

and we determine the mixing constants

$$u^*(\frac{k_1}{n}, \frac{k_2}{n}..)$$

Special case of Dirichlet: Beta distribution

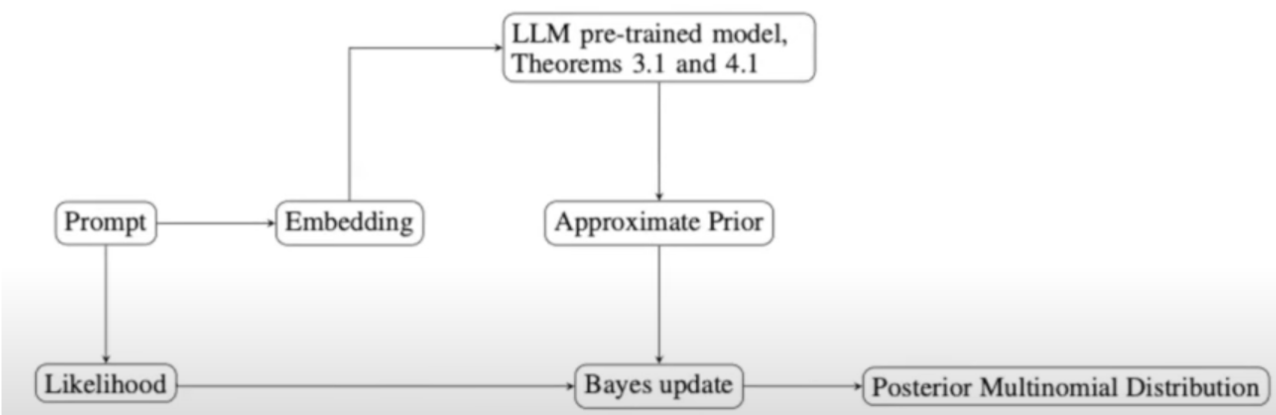
Conceptual multinomial distribution generation process

Bayes theorem:

$$\text{Posterior} = \frac{\text{Prior } X \text{ likelihood}}{\text{Evidence}}$$

Given prompt(Eg: The cat sat on the) (this is the "likelihood")

- Convert to embedding
- From the LLM pre-trained LLM, Using the continuity and universal representation theorem, Find embedding close to the prompt. This is the "approximate prior" for the bayesian model
- Model looks at the promt again. This is the "likelihood"
- Using this, performs the "Bayes update" and computes the posterior multinomial distribution.
- Posterior is used to generate the next token. This is repeated for every token.



In-context learning is like a Bayesian update mechanism. It can be some other mechanism, but it displays abilities consistent with Bayesian Updating. Occam's Razor :)

An exercise: In Context Learning

- Pick the most difficult case: Semantically unrelated label in context learning

- Let the prior or pre-trained label for a prompt be "A"
- Let the distribution of labels be a Beta prior, with two labels A and B

$$Beta(\alpha_a, \beta_b)$$

  - If the training data is primarily label A with the rare occurrence of B, then we will have

$$\alpha_a >> \beta_b$$

- We produce "n" samples of a prompt X, and labels A and B

Bayesian update

- Now conside ICL. Here, we are replacing A by B in n prompts
- Thus we have $x_b = \text{n prompts of B}$ and $x_a = 0 \text{ prompts of A}$
- The posterior probabilities p_a and p_b with n samples of label B for the prompts is given by

$$E(p_a|x_a, x_b) = \alpha_a/(\alpha_a + \beta_b + n)$$

$$E(p_b|x_a, x_b) = (\beta_b + n)/(\alpha_A + \beta_B + n)$$

Two cases with different α_A and β_B, maintaining ratio

n	$E(p_A|n)$	$E(p_B|n)$
0	0.968	0.032
1	0.229	0.771
2	0.13	0.87
3	0.091	0.909

When α_A and β_B are *small*, the probabilities **flip** with only **3** samples.

Table 1: Behavior of $E(p_A|n), E(p_A|n)$ with n prompts and $\alpha = 0.3\, \beta = 0.01$

n	$E(p_A|n)$	$E(p_B|n)$
0	0.968	0.032
1	0.732	0.268
2	0.588	0.412
3	0.492	0.508

With larger α_A and β_B probabilities are slower to change

Table 2: Behavior of $E(p_A|n), E(p_A|n)$ with n prompts and $\alpha = 3\, \beta = 0.1$

Even if we had a small model, but had a larger context size, we could've flipped this probability with enough examples.

Interpretation of α_A and β_B and generalization

- The parameters α_A and β_B directly correspond to the size of the network(and the training data)
    - The larger the network(parametr space), the smaller are the individual values of these parameters
    - With diverse training data,a the probabilities get scattered across many more labels resulting in smaller α_A and β_B
    - In-context-learning "emerges" in larger networks because fewer examples are needed to move the probabilities via Bayesian updating.
    - The examples and intuition can be generalized to any multimodal distribution by the universal representation theorem.

Implications

- Some kind of bayesian switch is getting turned on in these networks to enabled optimal predictions
- Embeddings play a key role, especially continuity of embeddings to multinomial distributions
- Given that other architectures like Mamba etc. are showing similar behaviour, Transformers/Attention may not be the key => next token prediction is the key.
- Model explains phonomena like Chain of Thought reasoning(good priors exist for component steps in training data)
    - Smaller steps are likely seen in the training data before, leading to better overall results
    - When sampling tokens, if you pick the token in such a way that the entropy reduces, it means that the model is becoming more confident.
    - Model has likely seen smaller steps, hence the entropy of selecting these tokens will be low, thereby leading to higher confidence, and a better final result.