IS813: Gen AI - Implementation

Mohannad Elhamod

BOSTON UNIVERSITY

Boston University Questrom School of Business

Neural Nets in Language Modeling

Continued…

BOSTON UNIVERSITY

Fast Forward...

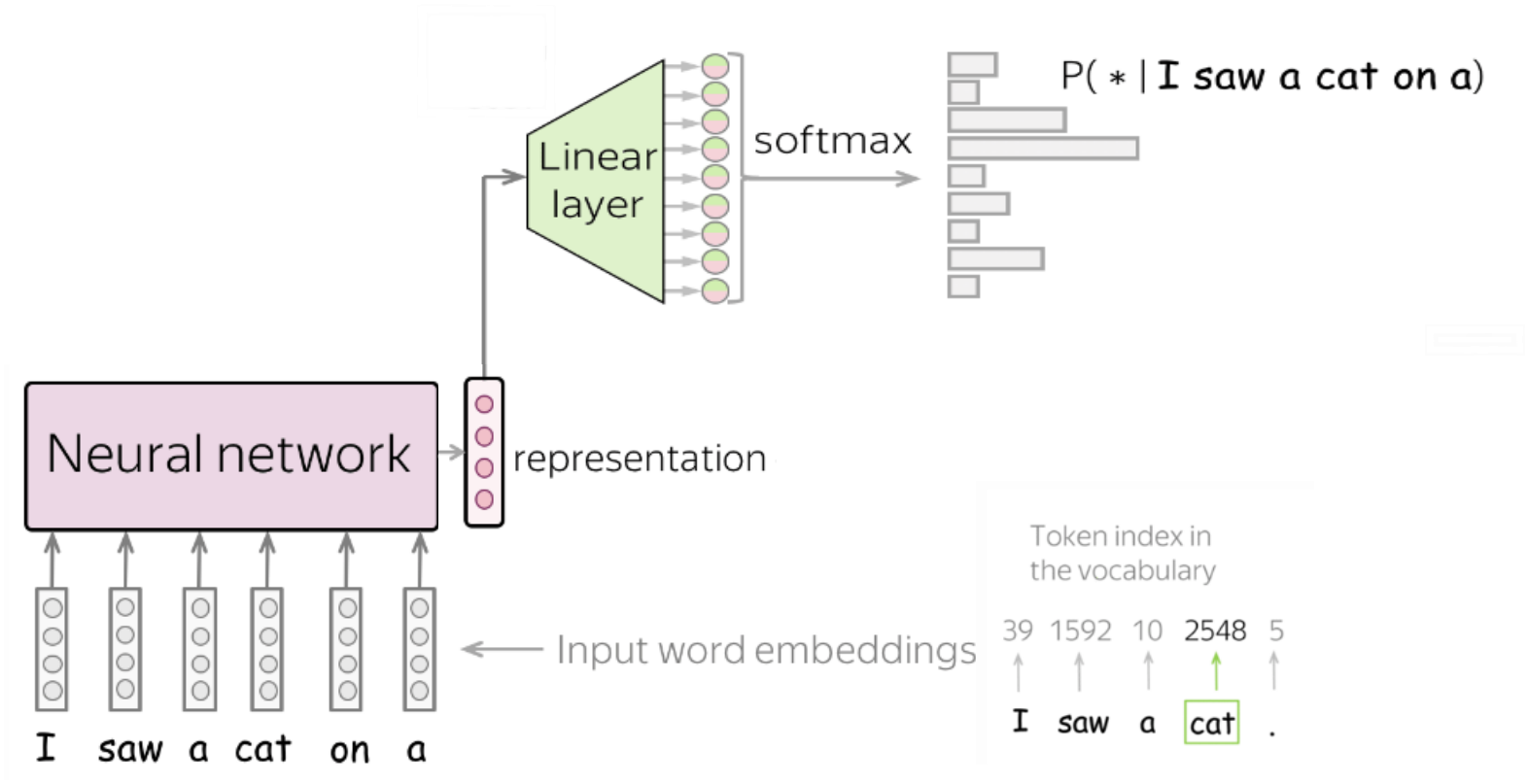
As neural networks arrived at the scene, they were utilized for language modeling.

- N-grams look for exact prefixes, which is limiting…
- However, neural networks can learn more interesting relationships between the words.

Example: All humans are mortal. Socrates is a human. Therefore,

Socrates is mortal.

General Model Architecture



Lena-voita

Can you see any issue with inputting words in an NN?

What is an embedding?

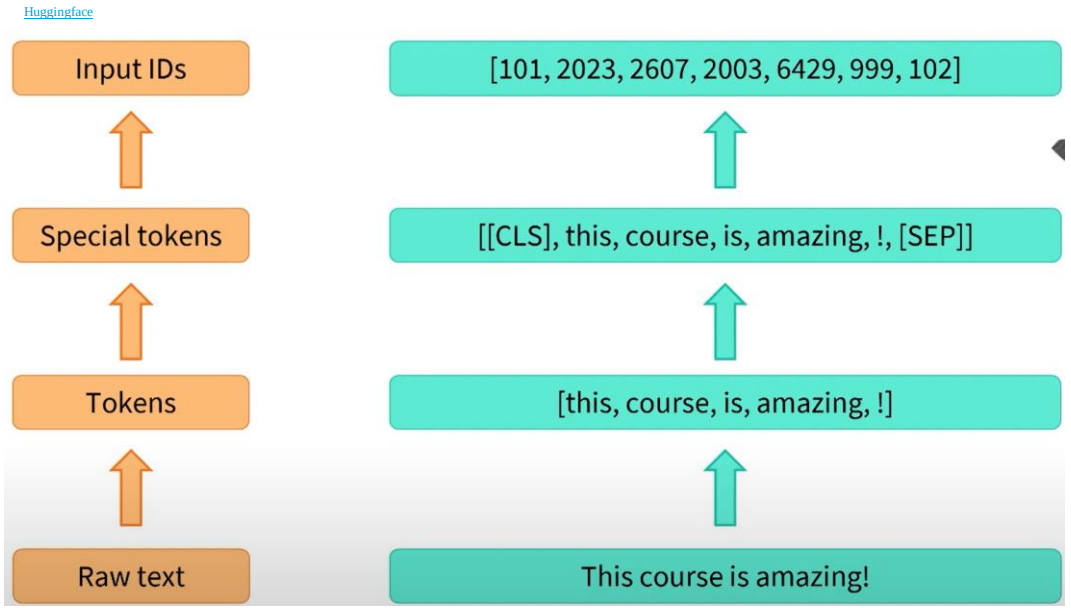
- It is the numeric representation of data.
- Example for images.

Word Embeddings

- We ideally want related words (i.e., *similar* meanings) to have smaller distances.
- Demo
- Examples:
  1. Word2Vec (Google)
  2. GloVe (Stanford)
  3. Train your own!

Tokenization

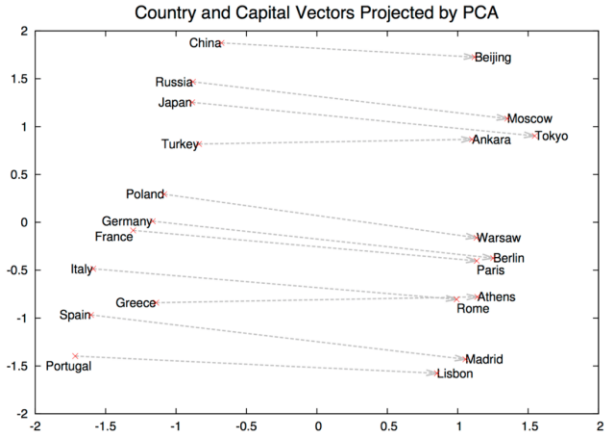
- Computers only understand numbers.
- We need to convert the text into tokens (e.g., words).
- Each token can then be represented as a number.



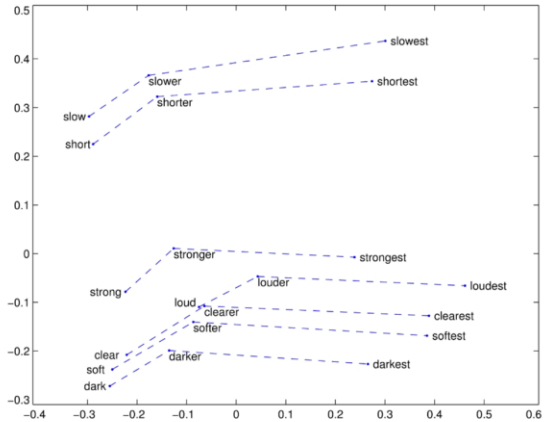
Word Embeddings

Since word embeddings carry *meaning*, certain directions in their space carry certain significance:

- Demo (dimensionality)



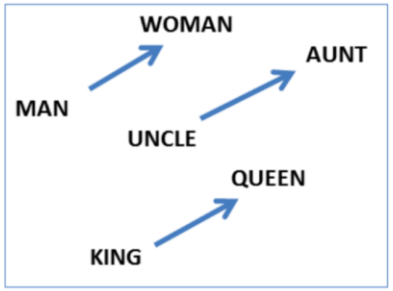
Word2Vec

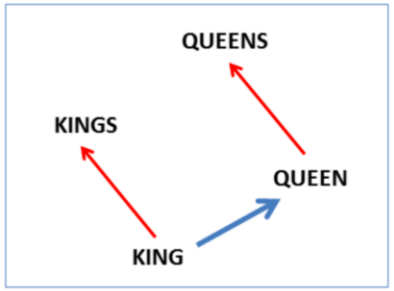


GloVe

semantic: $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$

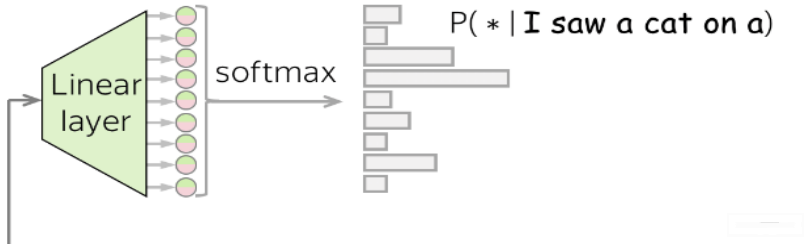
syntactic: $v(\text{kings}) - v(\text{king}) + v(\text{queen}) \approx v(\text{queens})$

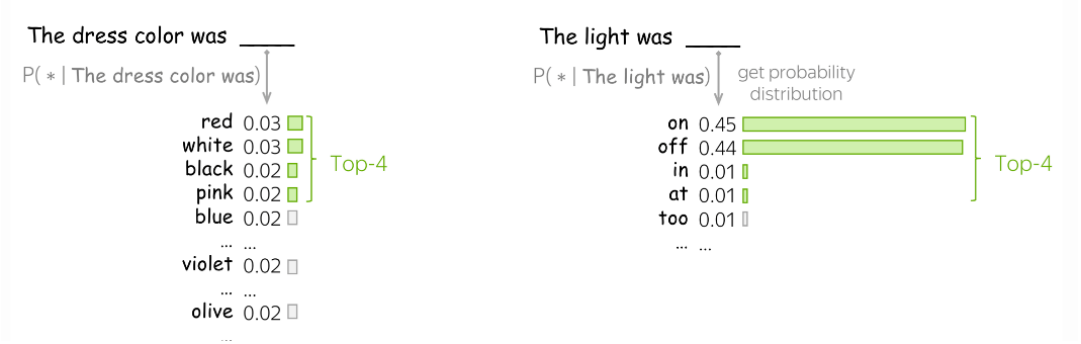


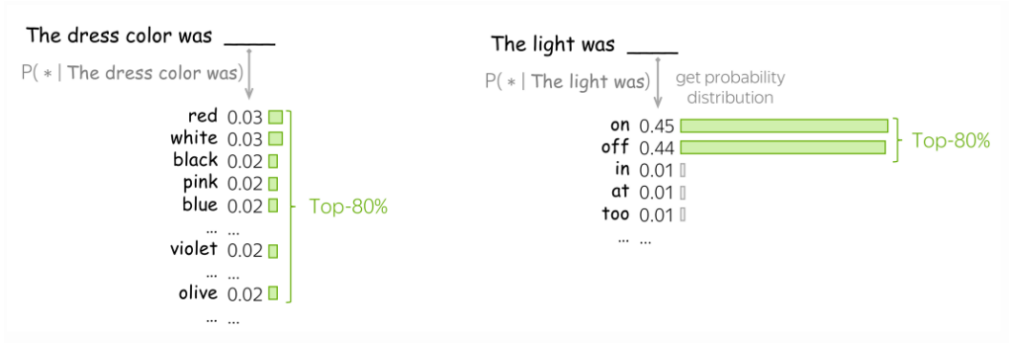


Sampling The Distribution

- Always take top probability?
  - That makes the model deterministic (no creativity).
- Alternative?
  - Top-k or top-p.



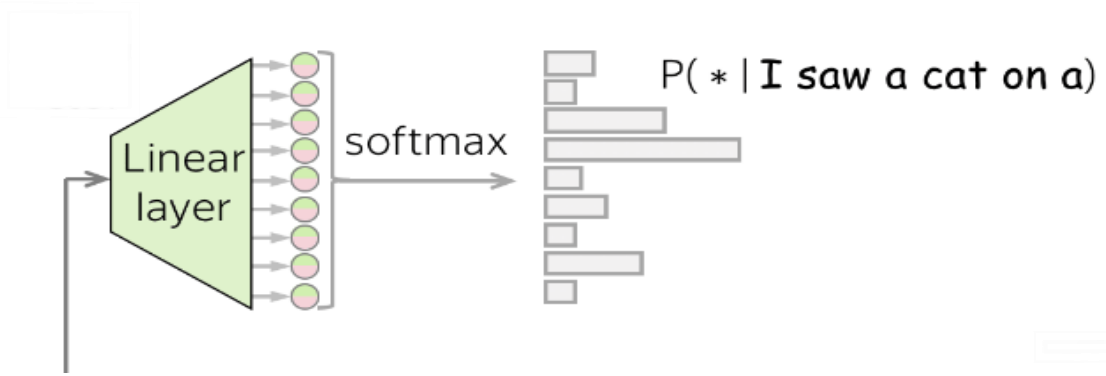


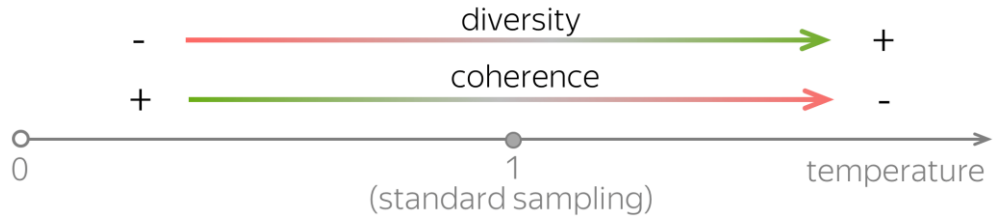


Lena-voita

Sampling The Distribution

- Some words have way higher probability than others.
- This can be manually tuned through temperature.
- Demo





Lena-voita

Measuring The Metric

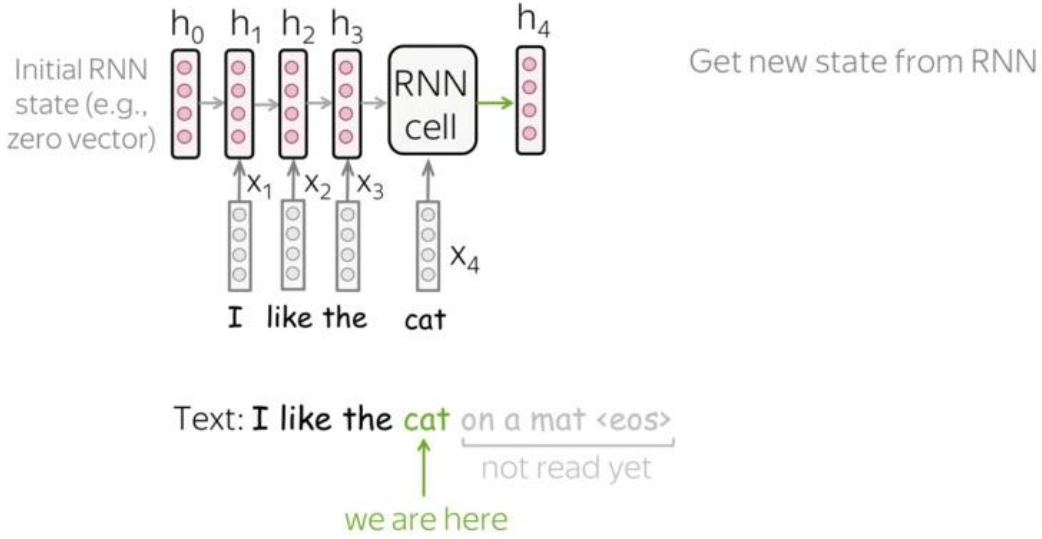
- What are we looking for?
  - A model that is not surprised by the new text it seen.
- We use perplexity.
  - Takes values between 1 and number of possible tokens.
  - Smaller is better.
  - Demo

Fast Forward...

- There exists many types of Neural Nets for language modeling:
  - CNNs
  - RNNs
  - LSTMs…

- Generally, Neural Nets learn an *embedding* that represents the *entire prefix* to predict the next word.

Lena-voita



Attention!

- These types of Neural Nets, however, suffered from various issues:
  - E.g., *catastrophic forgetting*, where earlier context in longer sentences tends to be forgotten.

- In 2015, *attention* in Neural Nets was invented:
  - It allowed models to attend to different parts of the sentence (instead of a single representation).

Published as a conference paper at ICLR 2015

NEURAL MACHINE TRANSLATION
BY JOINTLY LEARNING TO ALIGN AND TRANSLATE

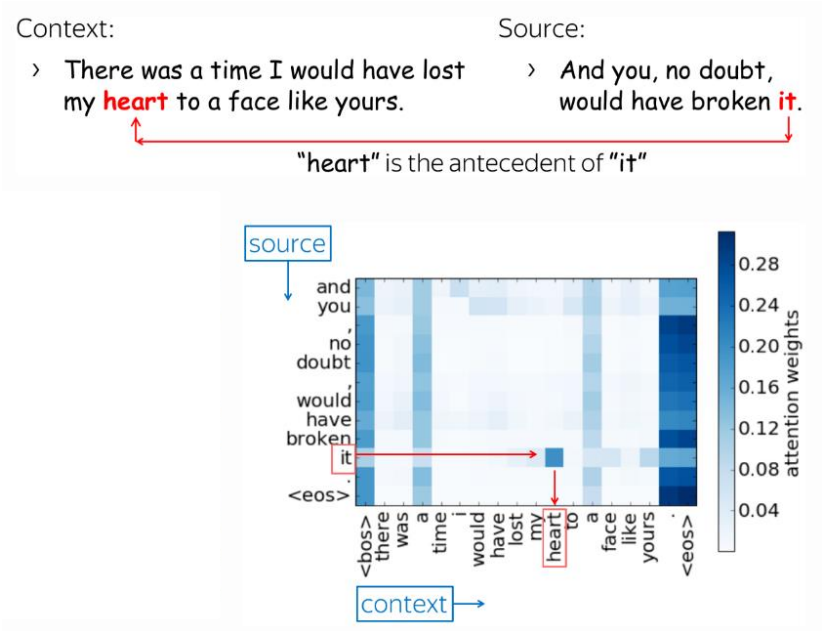
Dzmitry Bahdanau
Jacobs University Bremen, Germany

KyungHyun Cho **Yoshua Bengio***
Université de Montréal

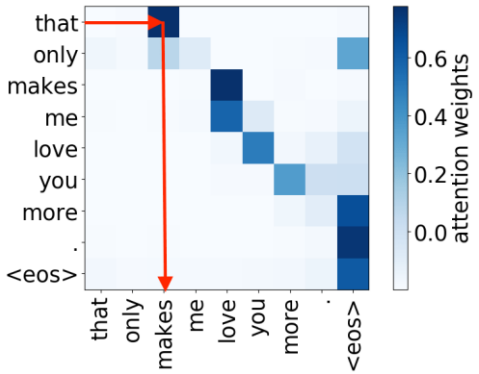
Lena-voita

Attention!

- Once each part has its own embedding, different types of *relationships* can be learned!



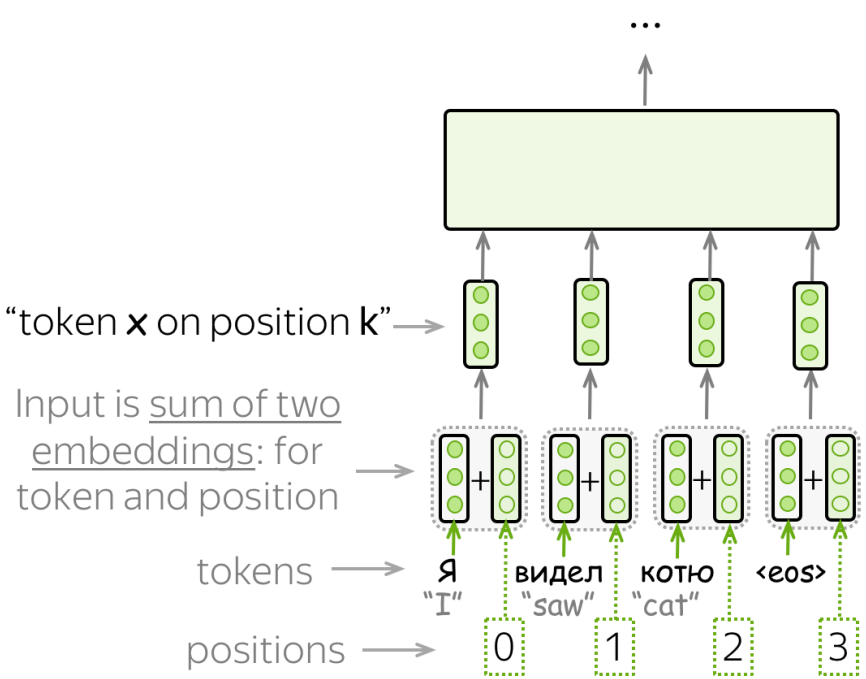
Lena-voita



Subject -> verb

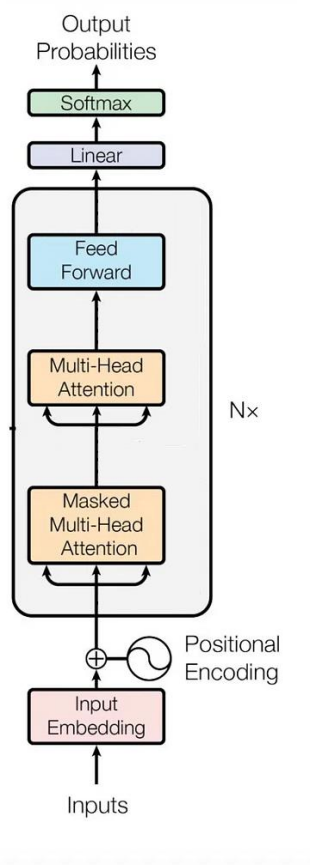
Order Matters: Positional Encoding!

- Since token embeddings do not contain information about the location of the word, they should be combined with a *positional encoding.*



Lena-voita

The Transformer is born!



Javinkarla

12 Jun 2017

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez*† University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* illia.polosukhin@gmail.com

Models in the wild

BOSTON UNIVERSITY

Model Types

We are not going to get into technical details, but certain models may be more fit for certain tasks:

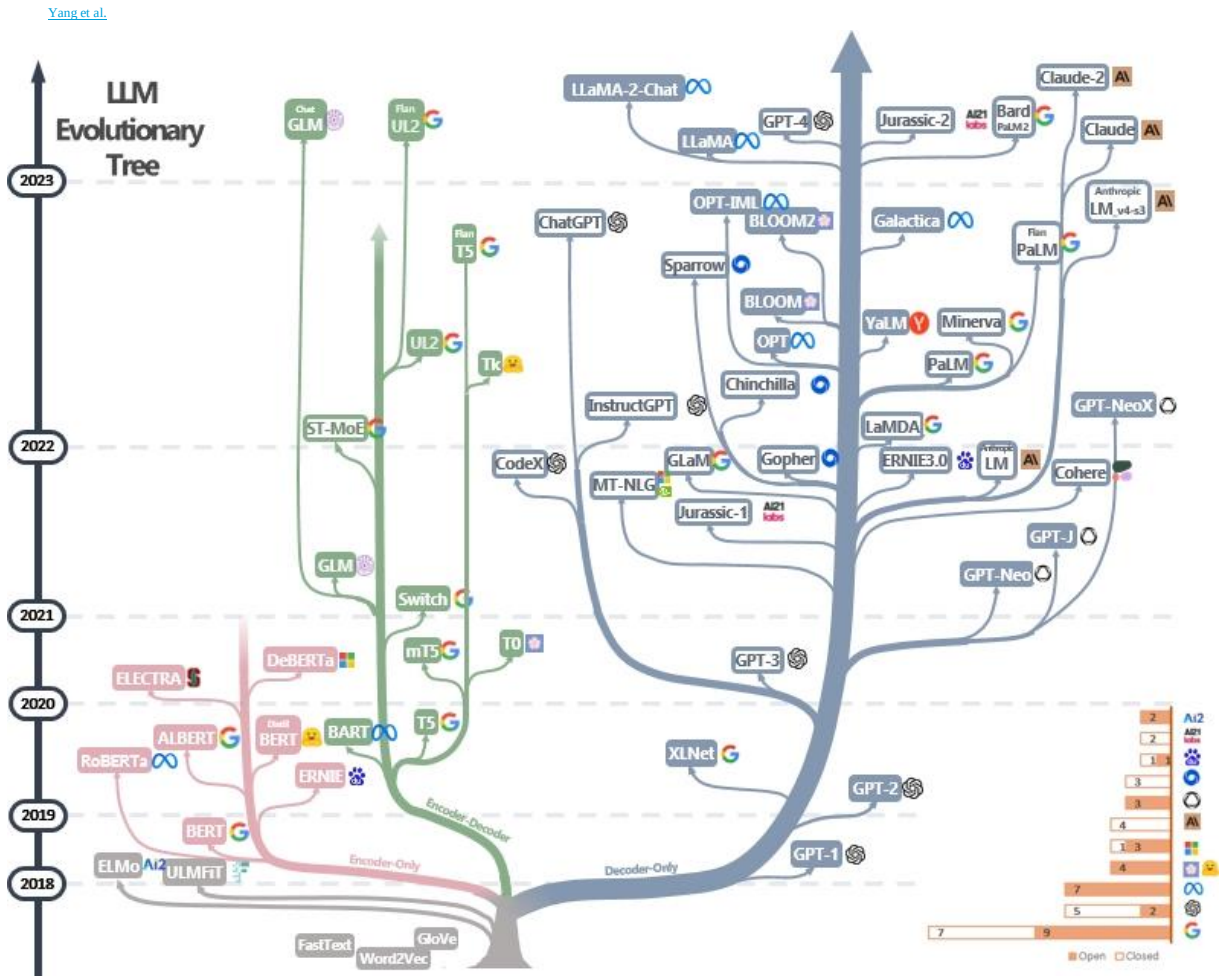
Model	Examples	Tasks
Encoder	ALBERT, BERT, DistilBERT, ELECTRA, RoBERTa	Sentence classification, named entity recognition, extractive question answering
Decoder	CTRL, GPT, GPT-2, Transformer XL	Text generation
Encoder-decoder	BART, T5, Marian, mBART	Summarization, translation, generative question answering

Javinkarla

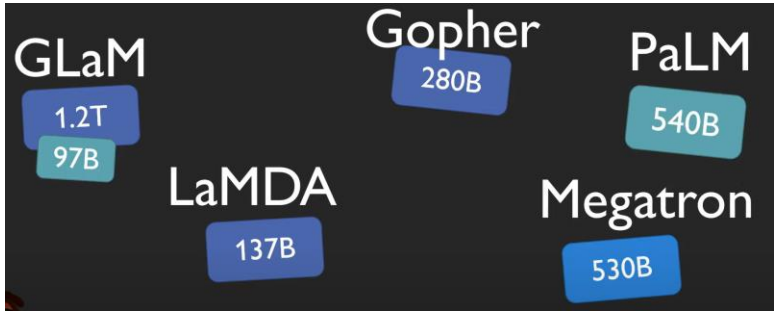
Why so many?

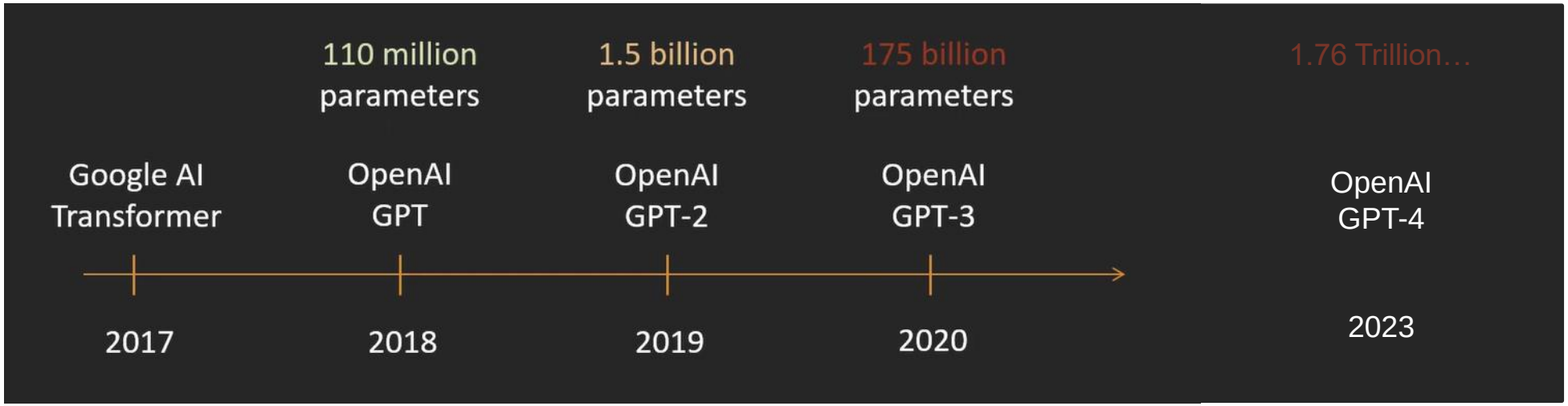
Where do the differences come from?

- Data.
- Model type and size.
- Hyperparameters (context size, embedding size,…).
- Training process (the cost function, fine-tuning, human feedback, etc.).



The GPT Evolution...





AI Coffee Beak with Letitia

Book Corpus
WebText

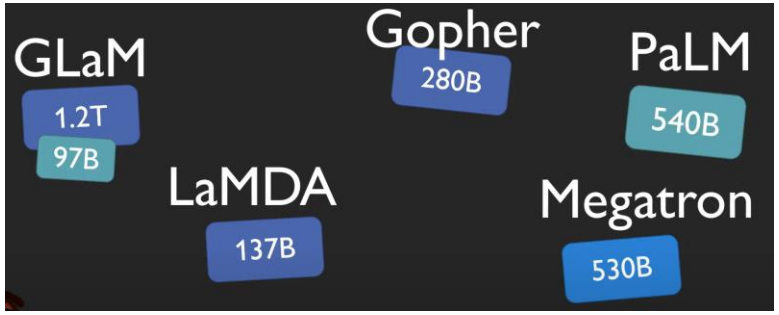
1,038 books (around 74M sentences and 1G words) of 16 different sub-genres (e.g., Romance, Historical, Adventure, etc.)

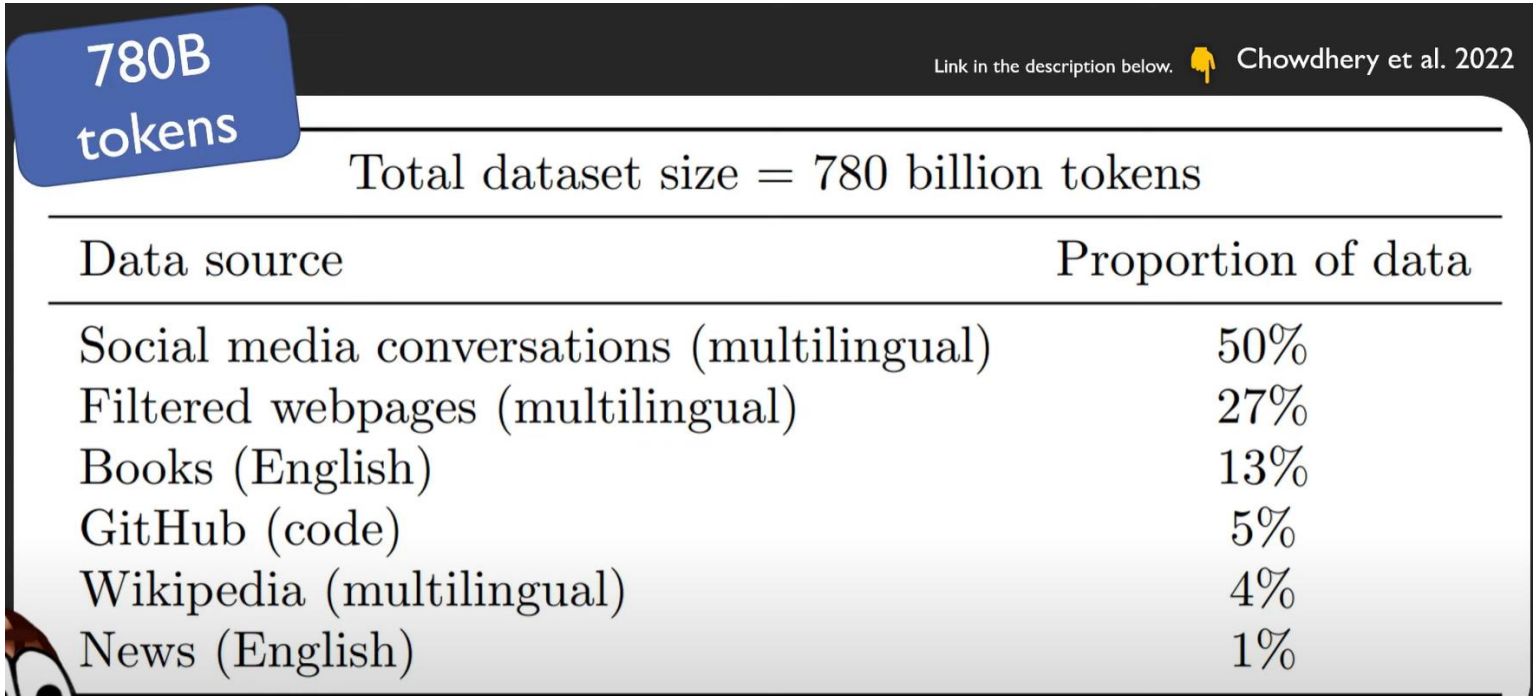
Common Crawl + ...

Over 240 billion pages. Petabytes of data.

????

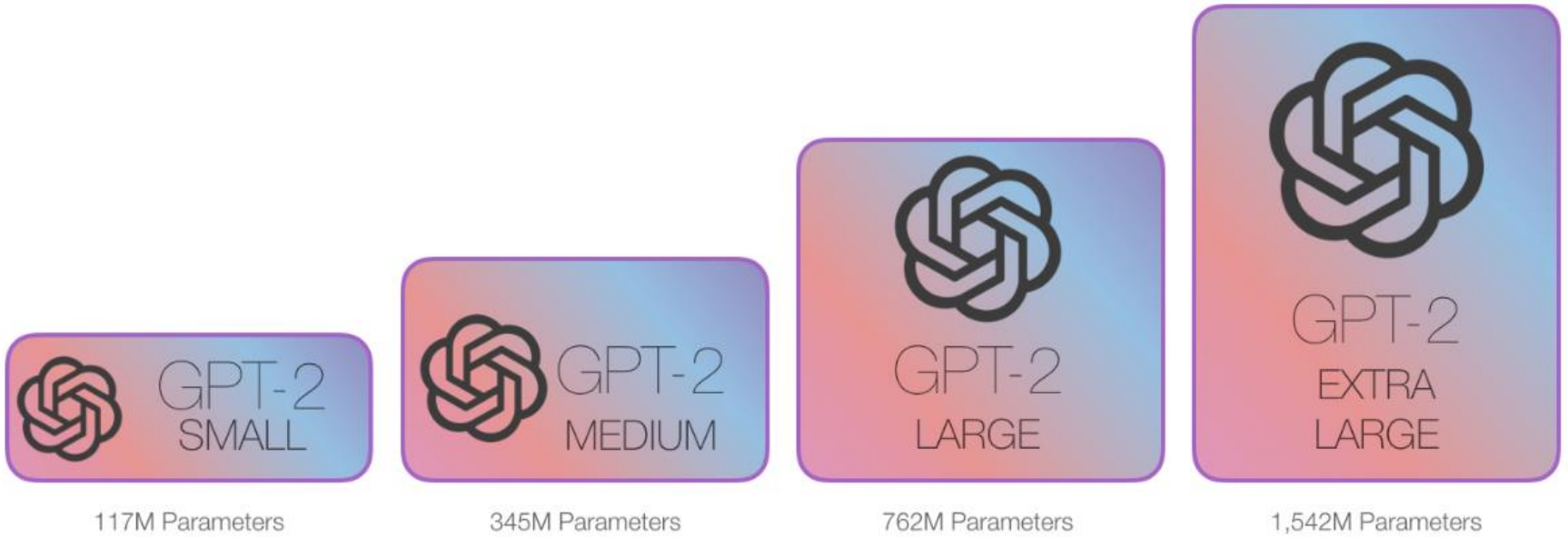
The GPT Evolution...





AI Coffee Beak with Letitia

Different model sizes



Jay Alammar

Exploring Your Options

- OpenAI model reference

- HuggingFace tasks
- HuggingFace models

How much training does it take?

2 example models

GPT-3 (2020)

50,257 vocabulary size
2048 context length
175B parameters
Trained on 300B tokens

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

Training: (rough order of magnitude to have in mind)

- O(1,000 - 10,000) V100 GPUs
- O(1) month of training
- O(1-10) $M

LLaMA (2023)

32,000 vocabulary size
2048 context length
65B parameters
Trained on 1-1.4T tokens

params	dimension	n heads	n layers	learning rate	batch size	n tokens
6.7B	4096	32	32	$3.0e^{-4}$	4M	1.0T
13.0B	5120	40	40	$3.0e^{-4}$	4M	1.0T
32.5B	6656	52	60	$1.5e^{-4}$	4M	1.4T
65.2B	8192	64	80	$1.5e^{-4}$	4M	1.4T

Table 2: **Model sizes, architectures, and optimization hyper-parameters.**

Training for 65B model:

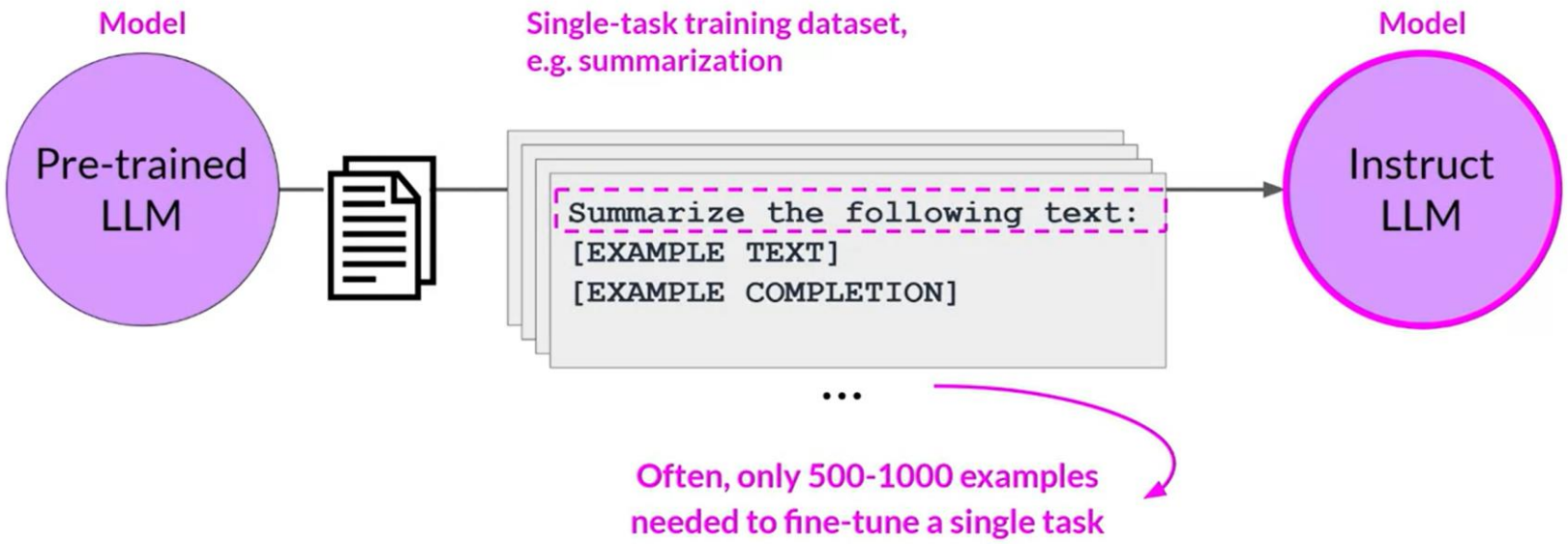
- 2,048 A100 GPUs
- 21 days of training
- $5M

[Language Models are Few-Shot Learners, OpenAI 2020]
[LLaMA: Open and Efficient Foundation Language Models, Meta AI 2023]

Pre-trained Models: Democratizing AI

- Most of us don’t have the expertise, data, or resources to train anything close to these impressive large models.
- Instead:
  - *Zero-shot Learning:* We can use open-source models out-of-the-box, even though they have never seen our data before.
  - *Transfer learning/Fine-Tuning:* Can be used as a base for further training (e.g., if the training data is non-public legal documents).

Example: Instruct LLMs



Coursera

In-Class Work

HuggingFace

BOSTON UNIVERSITY

Using APIs

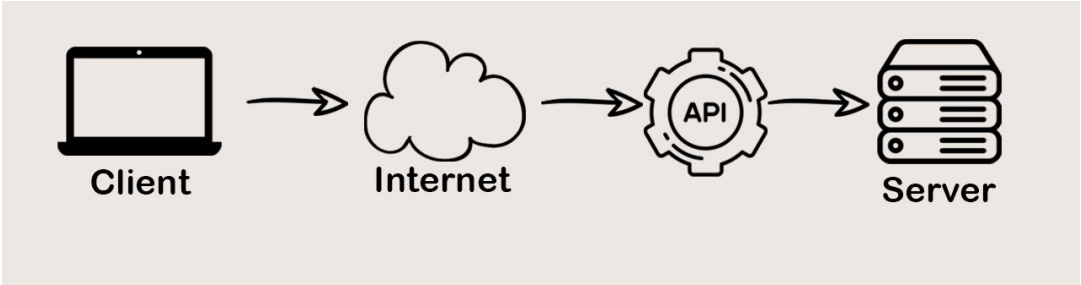
BOSTON UNIVERSITY

Boston University Questrom School of Business

What is an API?

An Application Programming Interface (API) establishes a *"contract"* or *"glue"* that allows multiple pieces of software to communicate.

- Standardizes software development.
- Allows integration with online services.
- Is generally well-documented.



CodeWithMazn

LLM APIs

APIs that provide access to LLMs:

- You send a query/request (e.g., a prompt or a sentence)
- You get a response (e.g., a summary or a translation)

Why LLM APIs (vs. in-house)?

- Pros?
  - Less memory and compute resources needed.
  - Abstraction in terms of maintenance and updates.
  - Security?

- Cons?
  - Less control.
  - Maybe slower.
  - Cost.
  - Privacy.

OpenAI API

- Documentation
- Playground
- Guide (notebook)
- Pricing

Resources

- Meaning and calculation of perplexity.
- Video: LLMs vs The Brain
- Video: Deciding which pre-trained model to fine-tune