ChatGPT and Friends

Bruno Gonçalves

graphs4sci.substack.com

https://github.com/DataForScience/ChatGPT

Question

https://github.com/DataForScience/ChatGPT

- What's your job title?
    - Data Scientist
    - Statistician
    - Data Engineer
    - Researcher
    - Business Analyst
    - Software Engineer
    - Other

Question

https://github.com/DataForScience/ChatGPT

- How experienced are you in Python?
    - Beginner (<1 year)
    - Intermediate (1-5 years)
    - Expert (5+ years)

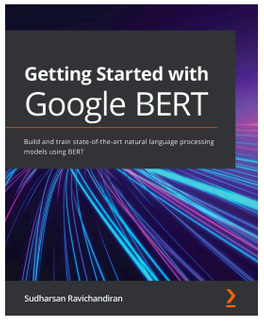
Question

https://github.com/DataForScience/ChatGPT

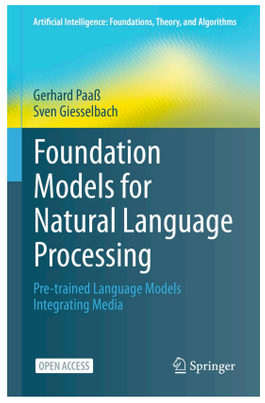
- How did you hear about this webinar?
    - O'Reilly Platform
    - Newsletter
    - data4sci.com Website
    - Previous event
    - Other?

References

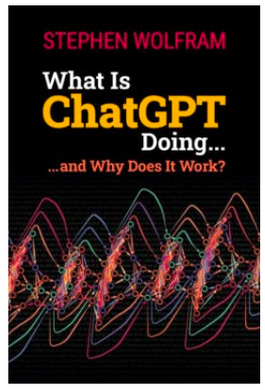
https://github.com/DataForScience/ChatGPT



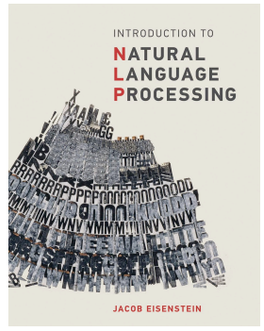
https://amzn.to/48u9qu5



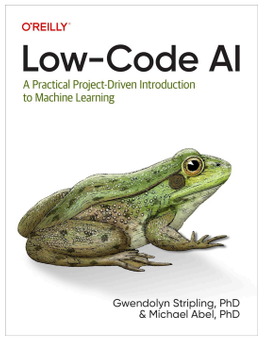
https://amzn.to/3P3qznx



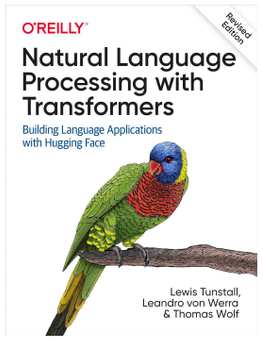
https://amzn.to/3LBRdBY



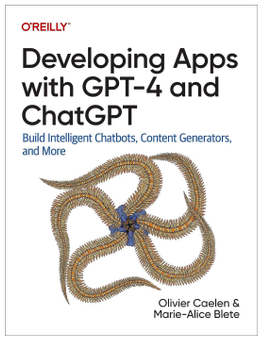
https://amzn.to/3ZMnTih



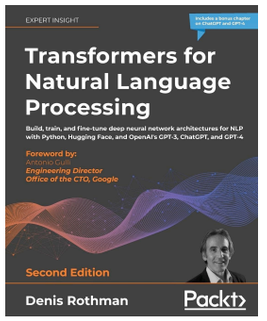
https://amzn.to/3UC7b4k



https://amzn.to/3UAaetS



https://amzn.to/3RHRkQa



https://amzn.to/46kOo02

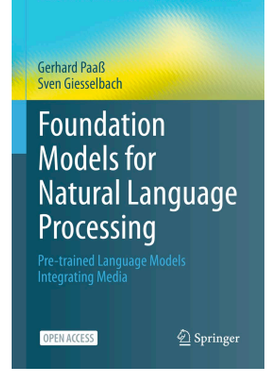
Table of Contents

1. Language Models

Language Models

- A foundational component of Natural Language Processing and AI.
- Software that predicts and generates human language based on patterns observed in training data.
- Applications to:
    - Machine translation
    - Summarization
    - Sentiment analysis
    - Chatbots and Virtual Assistants
    - Content Generation
    - etc…

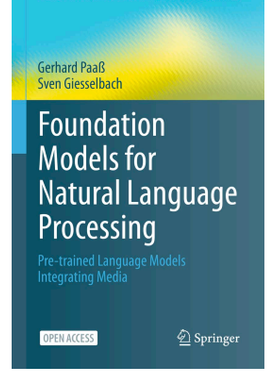
A Language Model is…



https://amzn.to/3P3qznx

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"
- Three common "flavors":
  - Linguistic - encode language as an explicit set of rules
  - Statistical - represents language as a set of word sequences and their respective probability
  - Word Embedding - compute a vector representation for each word in a vocabulary in such a way that the respective vector distance can estimate syntactical similarity between words

A Language Model is…



https://amzn.to/3P3qznx

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"
- Three common "flavors":
  - Linguistic - encode language as an explicit set of rules
  - **Statistical** - represents language as a set of word sequences and their respective probability
  - Word Embedding - compute a vector representation for each word in a vocabulary in such a way that the respective vector distance can estimate syntactical similarity between words

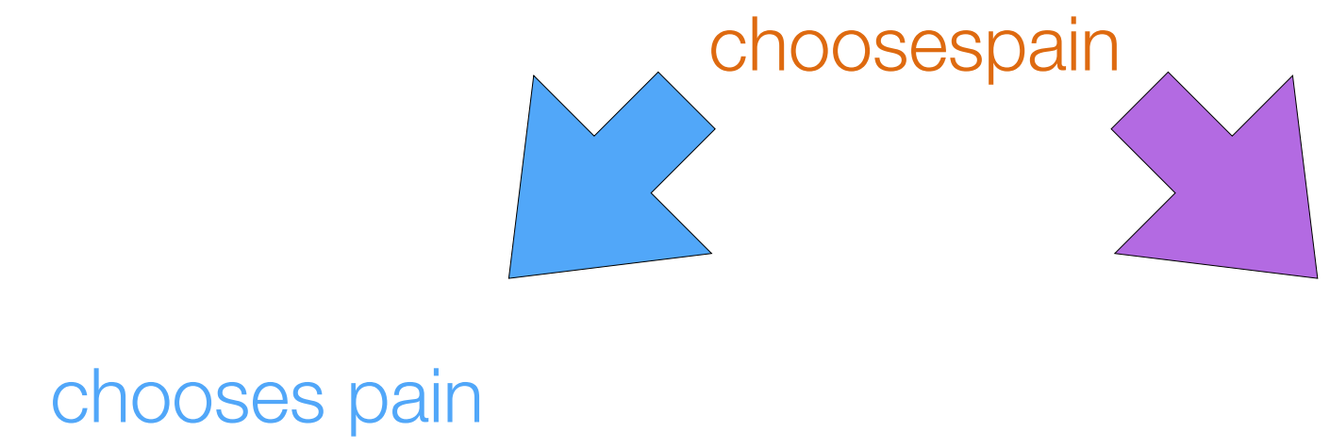
An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"

choosespain

An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"



An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"

chooses pain

$$P\left(chooses\right)P\left(pain\right)$$

choose spain

$$P\left(choose\right)P\left(spain\right)$$

An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"

choosespain

Probability that the word "spain" follows the word "choose"

chooses pain

$$P(chooses)\,P(pain)$$

1-gram

$$P(chooses)\,P(pain|chooses)$$

2-gram

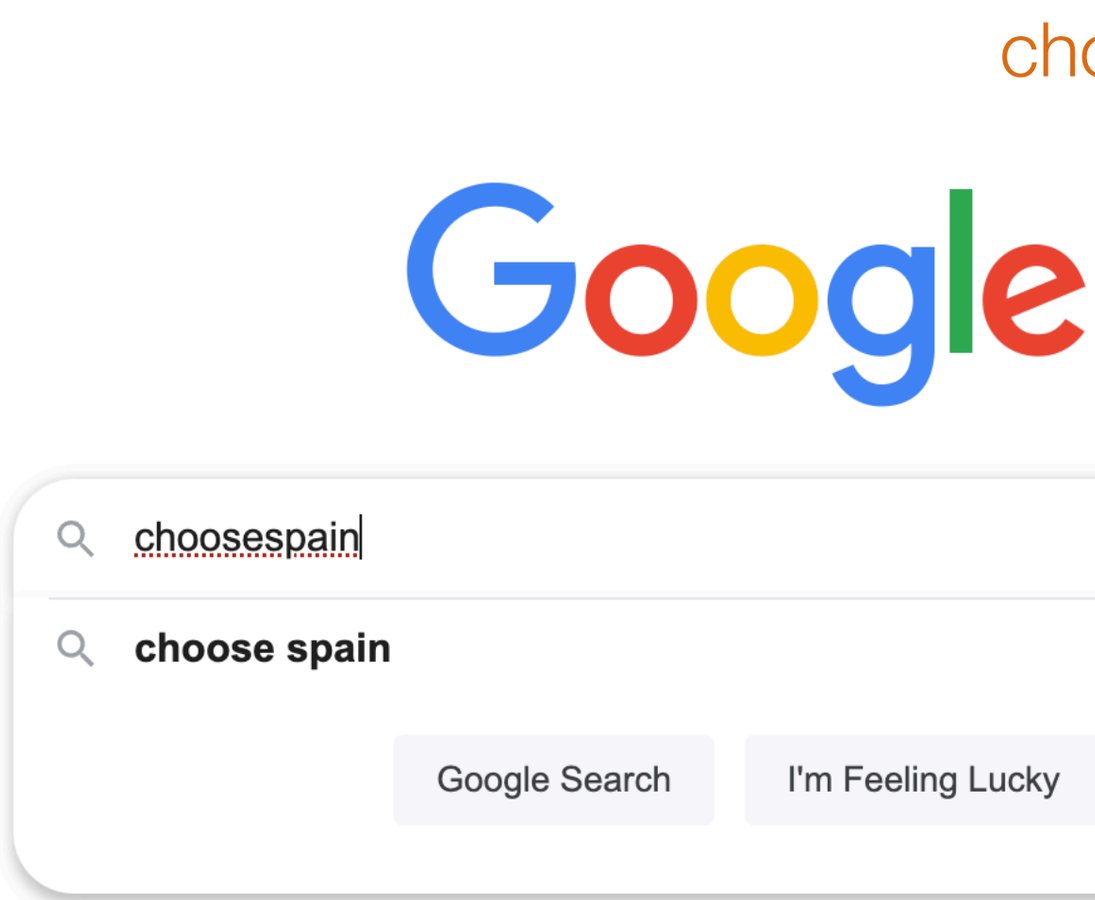
choose spain

$$P(choose)\,P(spain)$$

$$P(choose)\,P(spain|choose)$$

An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"



choosespain

Probability that the word "spain" follows the word "choose"

choose spain

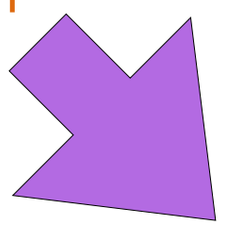
$$P(choose)\,P(spain)$$

$$P(choose)\,P(spain|choose)$$

An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"

choosespain



choose spain

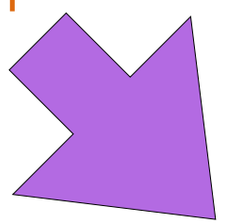
What are the most likely words to follow the word "choose"?

$$P\left(choose\right) P\left(spain \mid choose\right)$$

An example

- "A model that computes the joint distribution or the conditional probability of a Natural Language Text"

choosespain



choose spain

What are the most likely words to follow the word "choose"?

Select the next word base on the context of the previous one

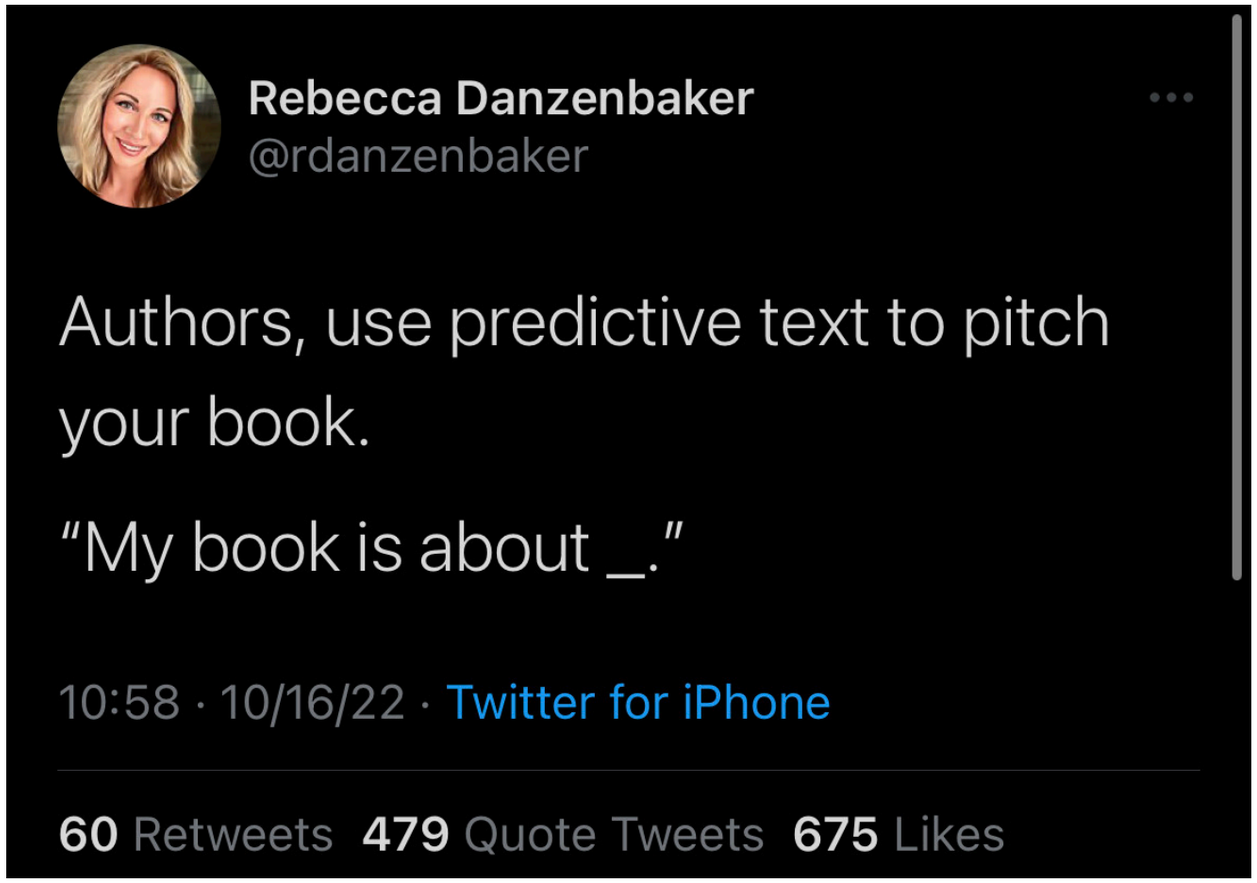
$$P\left(choose\right)P\left(spain\,|\,choose\right)$$

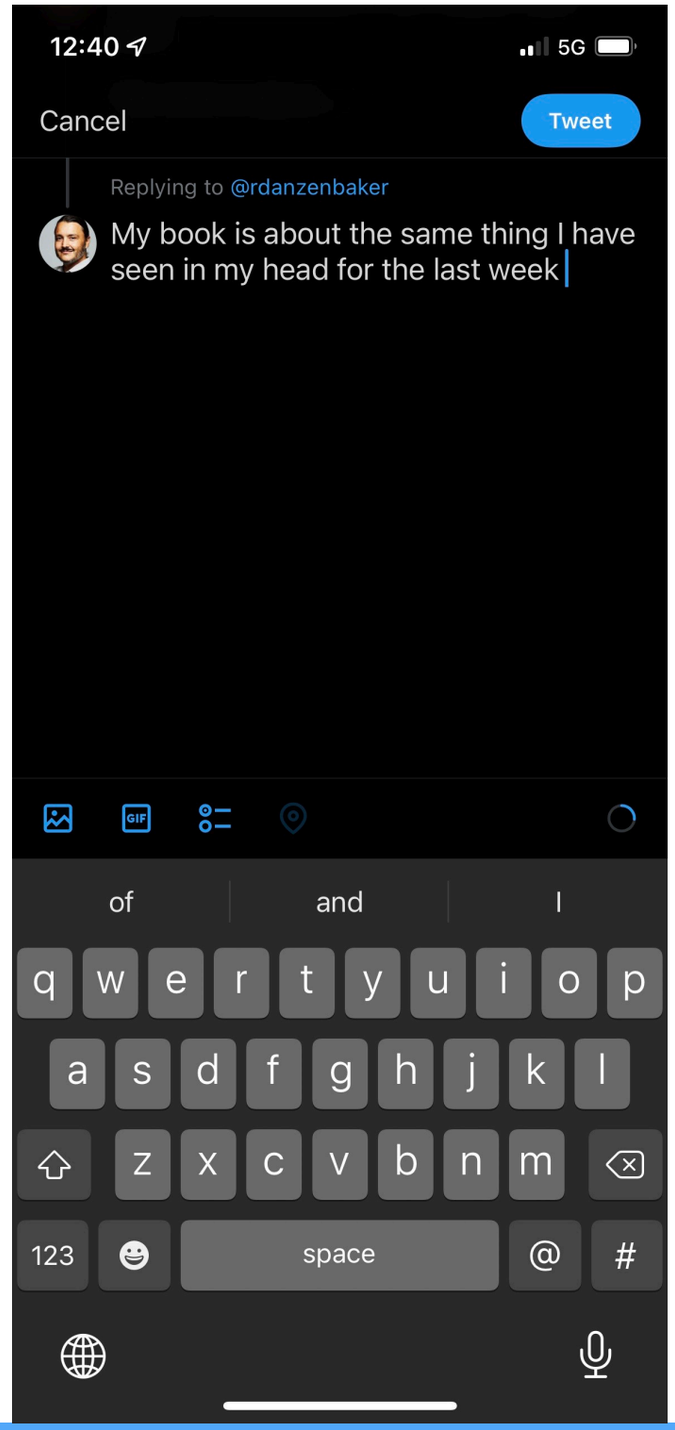
Language Models

- If you the predictive text feature in your digital devices, you're already familiar with this endless game of "guess the next word"...

Language Models

- If you the predictive text feature in your digital devices, you're already familiar with this endless game of "guess the next word"...

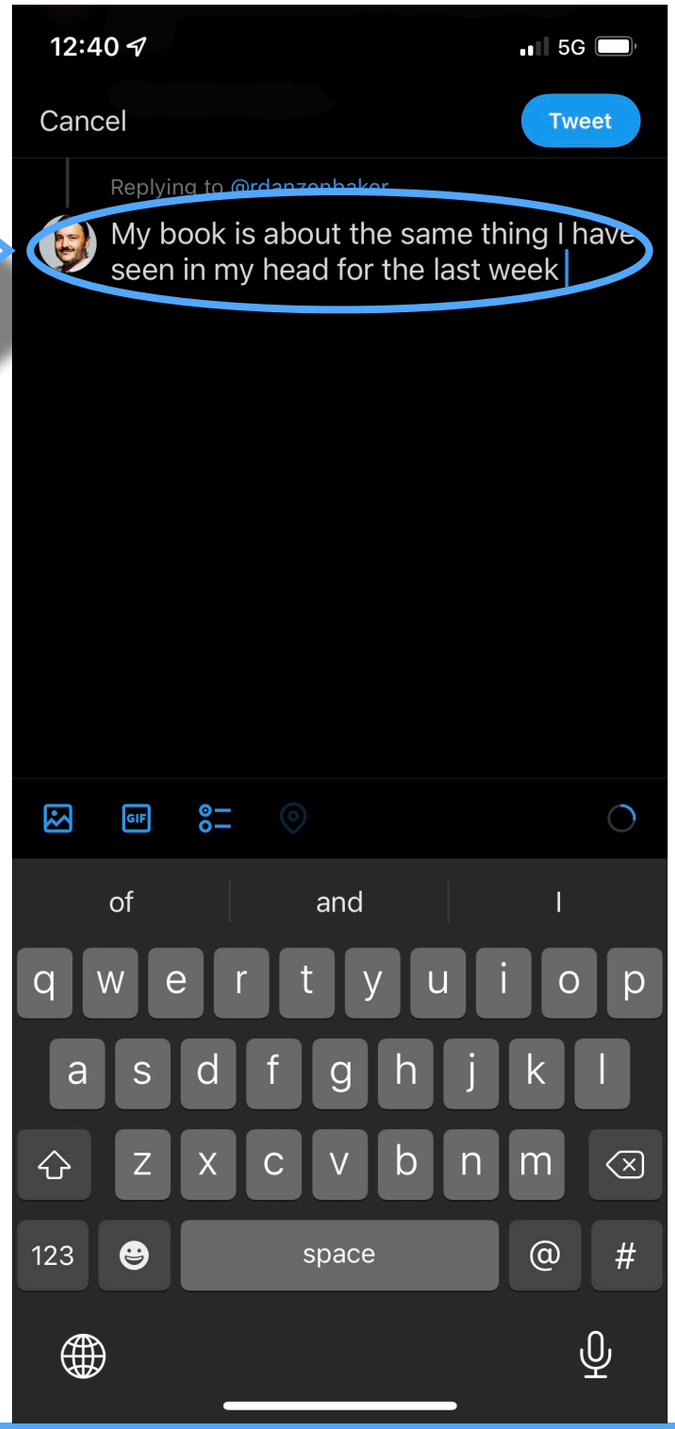




Language Models

- If you the predictive text feature in your digital devices, you're already familiar with this endless game of "guess the next word"...

Output



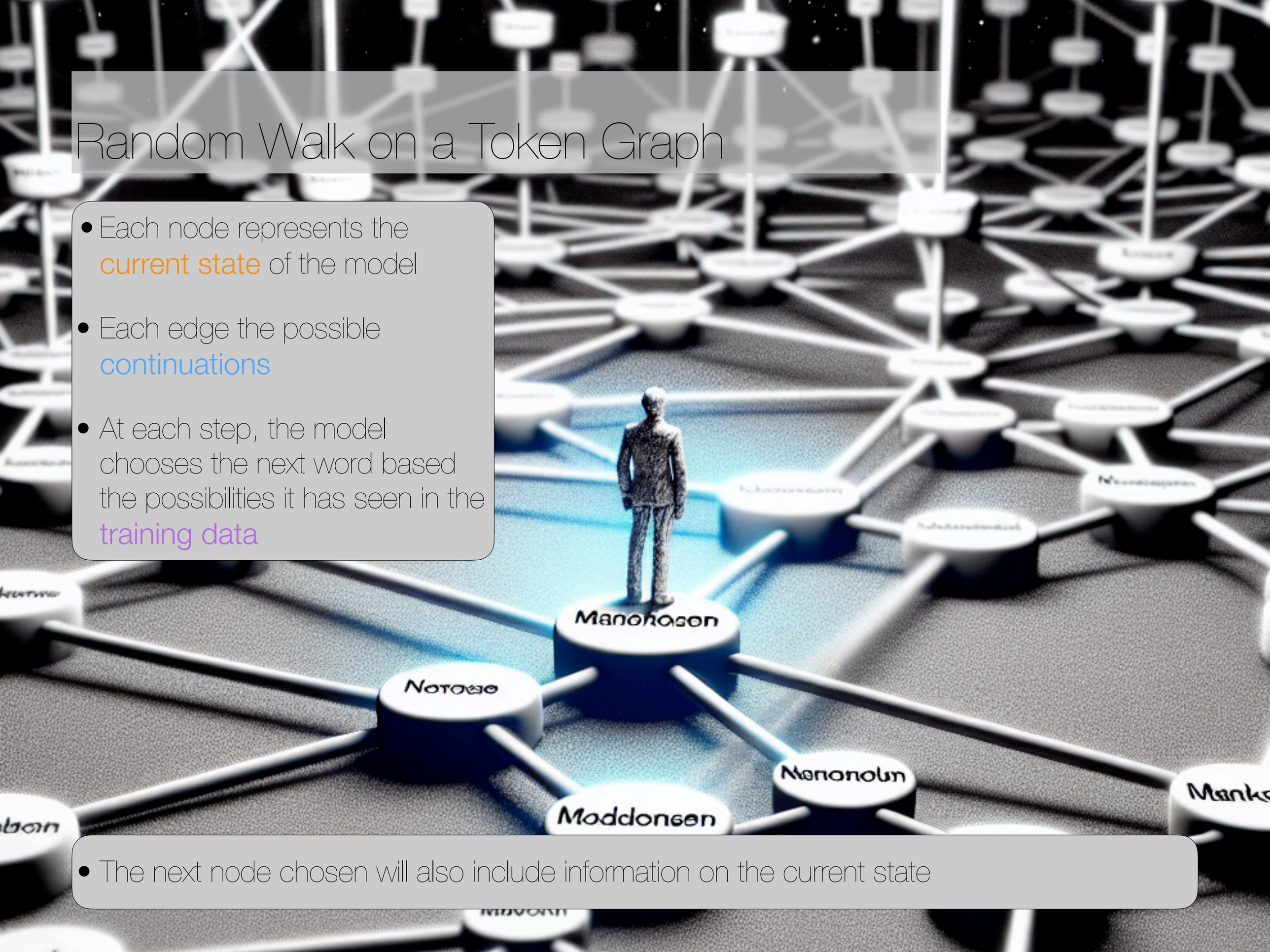
Random Walk on a Token Graph

Random Walk on a Token Graph

- Each node represents the current state of the model
- Each edge the possible continuations
- At each step, the model chooses the next word based the possibilities it has seen in the training data

Random Walk on a Token Graph

- Each node represents the current state of the model
- Each edge the possible continuations
- At each step, the model chooses the next word based the possibilities it has seen in the training data



- The next node chosen will also include information on the current state

Random Walk on a Token Graph

- Each node represents the current state of the model
- Each edge the possible continuations
- At each step, the model chooses the next word based the possibilities it has seen in the training data

- Large context windows and attention mechanisms allow the model to choose the best possible token (node) to generate at each step

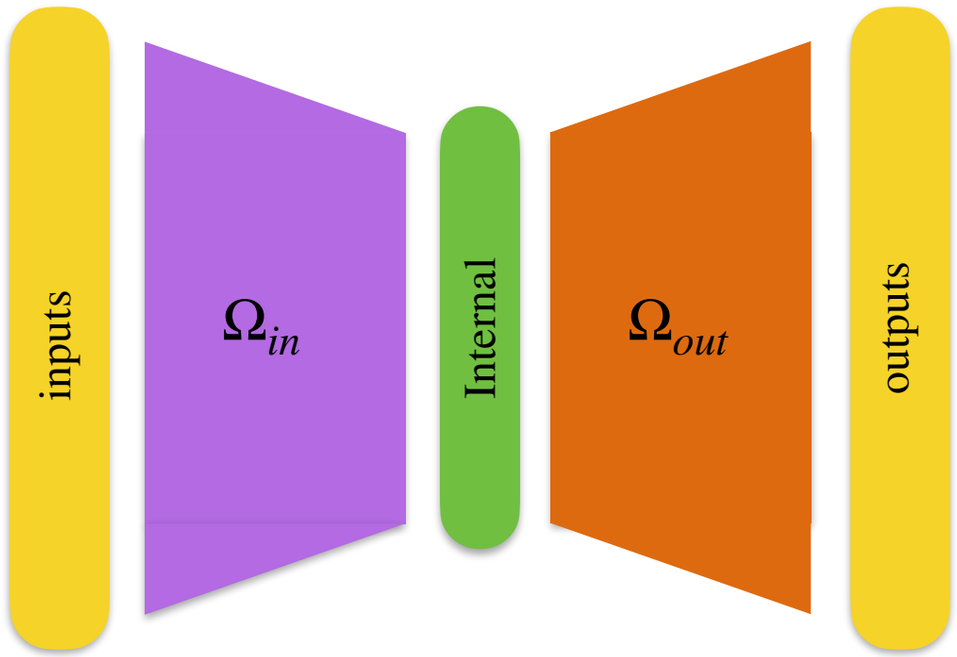
- The next node chosen will also include information on the current state

Language Models

https://github.com/DataForScience/ChatGPT

Encoder-Decoder

- Auto-Encoders use the same values for both inputs and outputs
- The Internal/hidden layer(s) have a smaller number of units than the input
- The fundamental idea is that the Network needs to learn an internal representation of its inputs that is smaller but from which it is still possible to reconstruct the input.



- Think of it as "zipping" and "unzipping" the input values. 

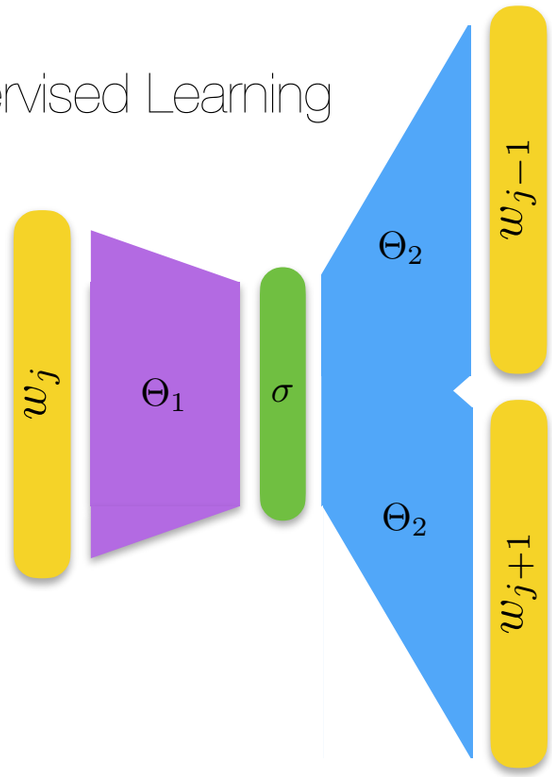
word2vec

Mikolov 2013

Skipgram

$$\max p\left(C \mid w\right)$$

Self-Supervised Learning



θ_1 word embeddings

θ_2 context embeddings

w_j one hot vector

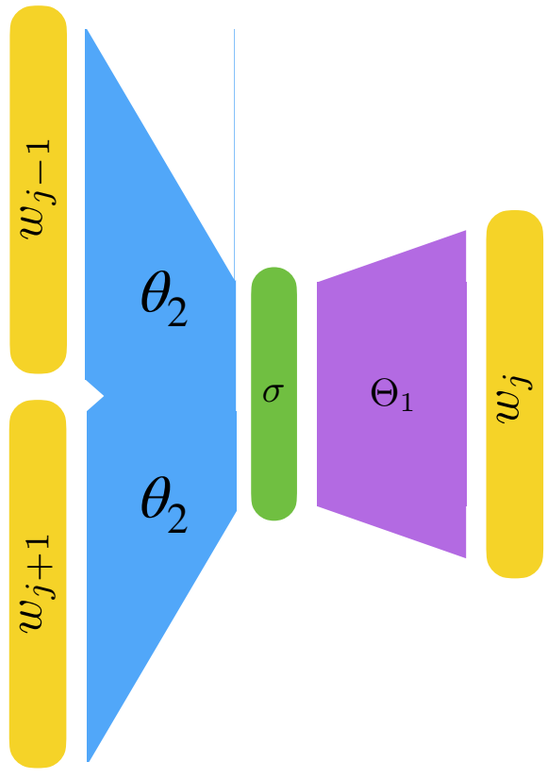
σ activation function

Word

Context

Continuous Bag of Words

$$\max p\left(w \mid C\right)$$



Context

Word

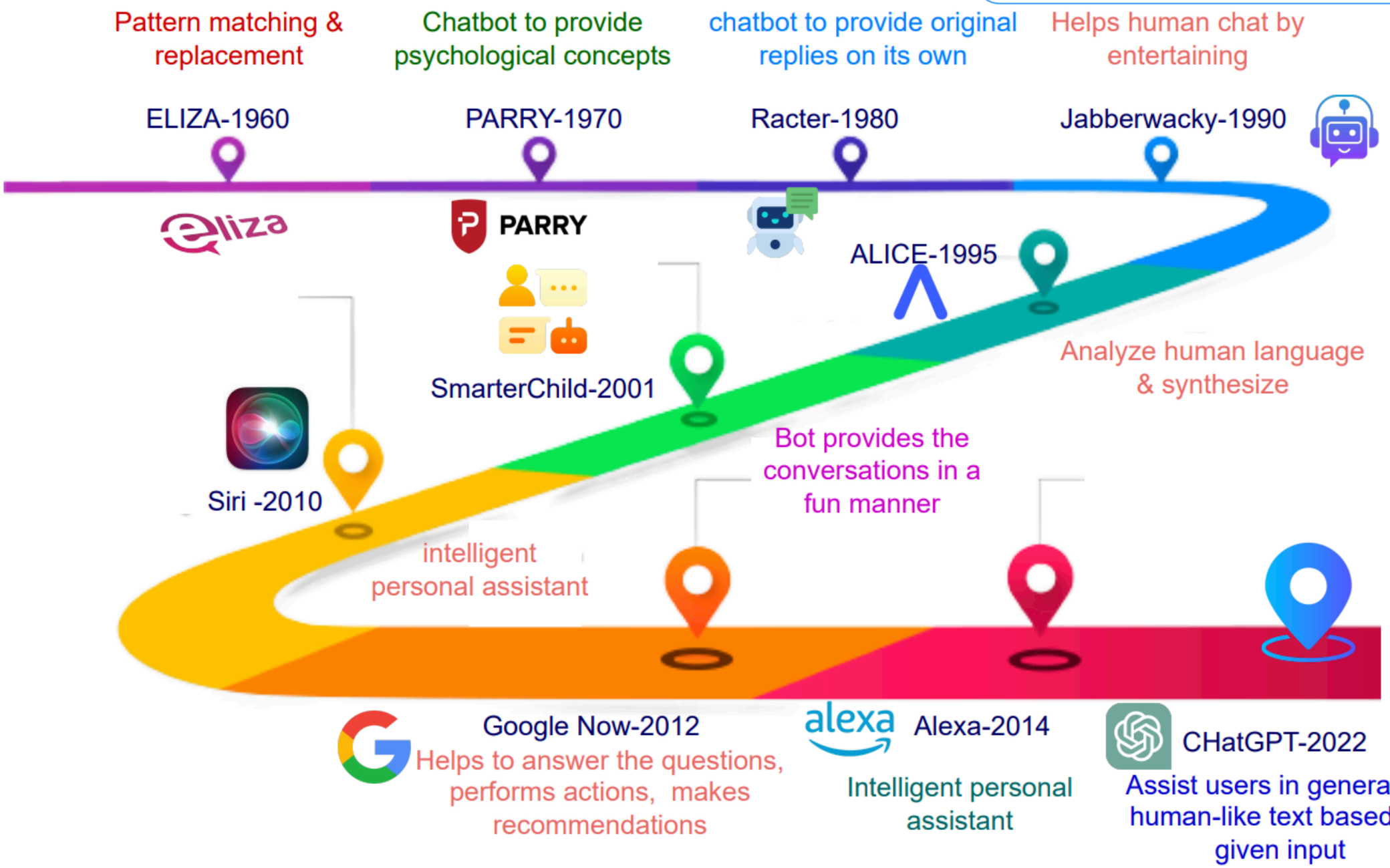
Word2vec

https://github.com/DataForScience/ChatGPT

2. Large Language Models

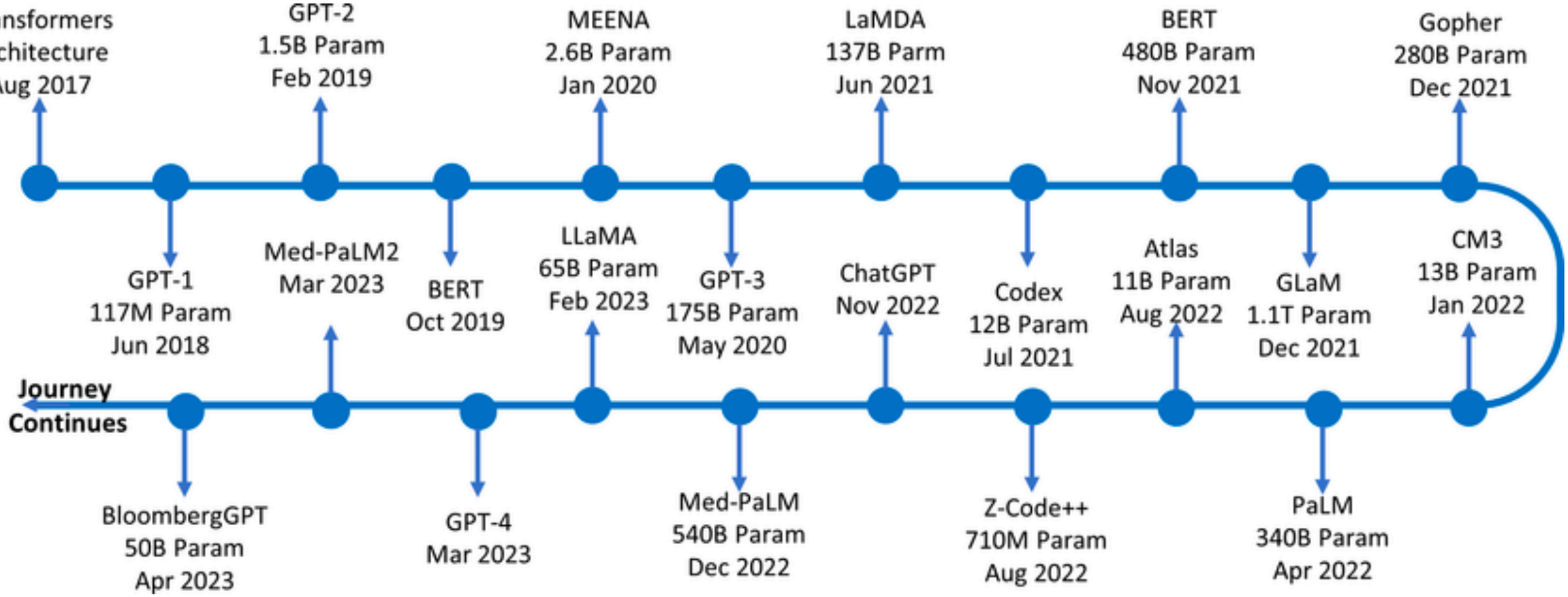
History of LLMs

arXiv:2305.10435 (2023)



History of LLMs

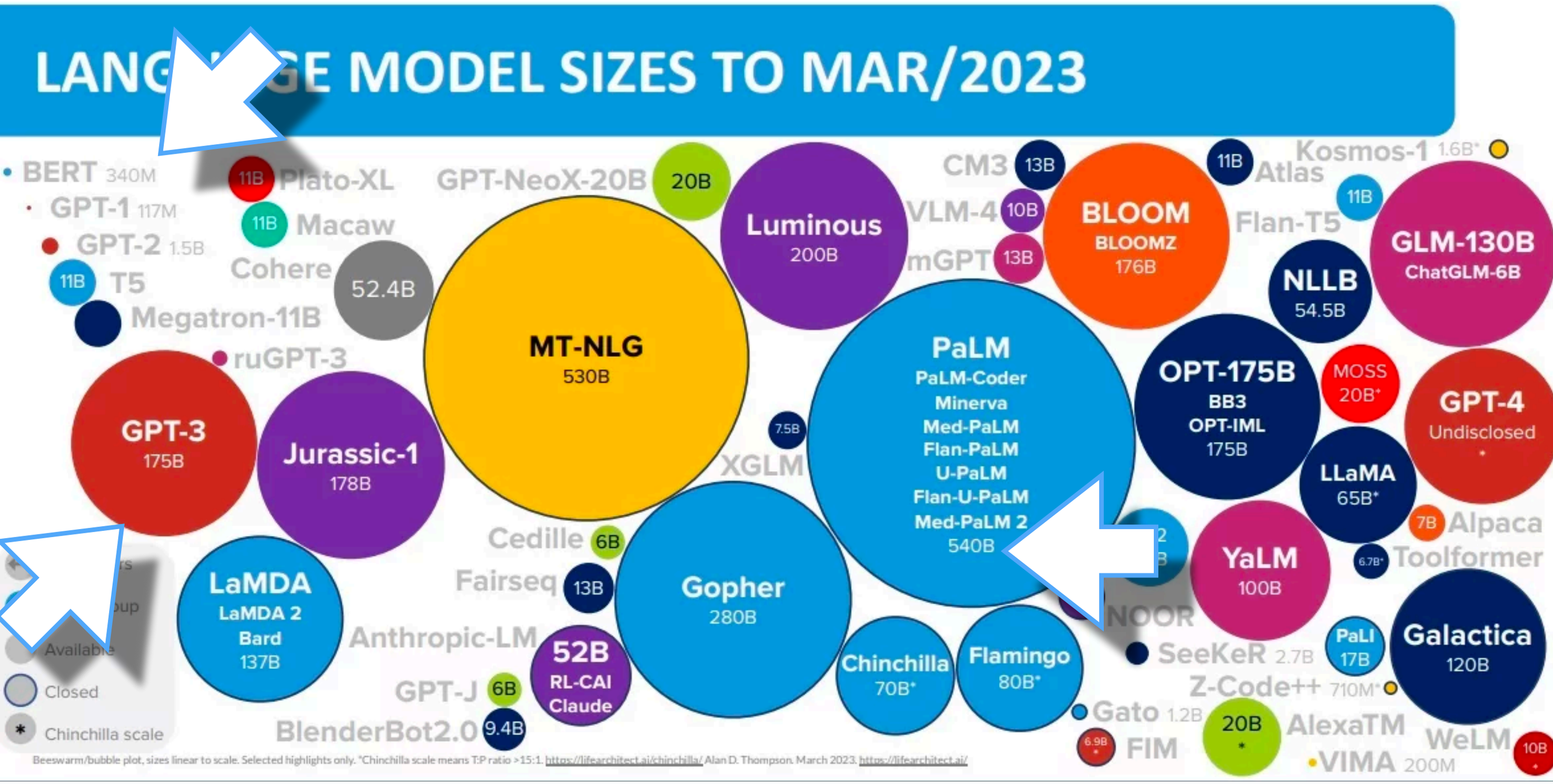
arXiv:2307.04251(2023)



Large Language Models

https://pureinsights.com/blog/2023/what-are-large-language-models-llms-search-and-ai-perspectives/

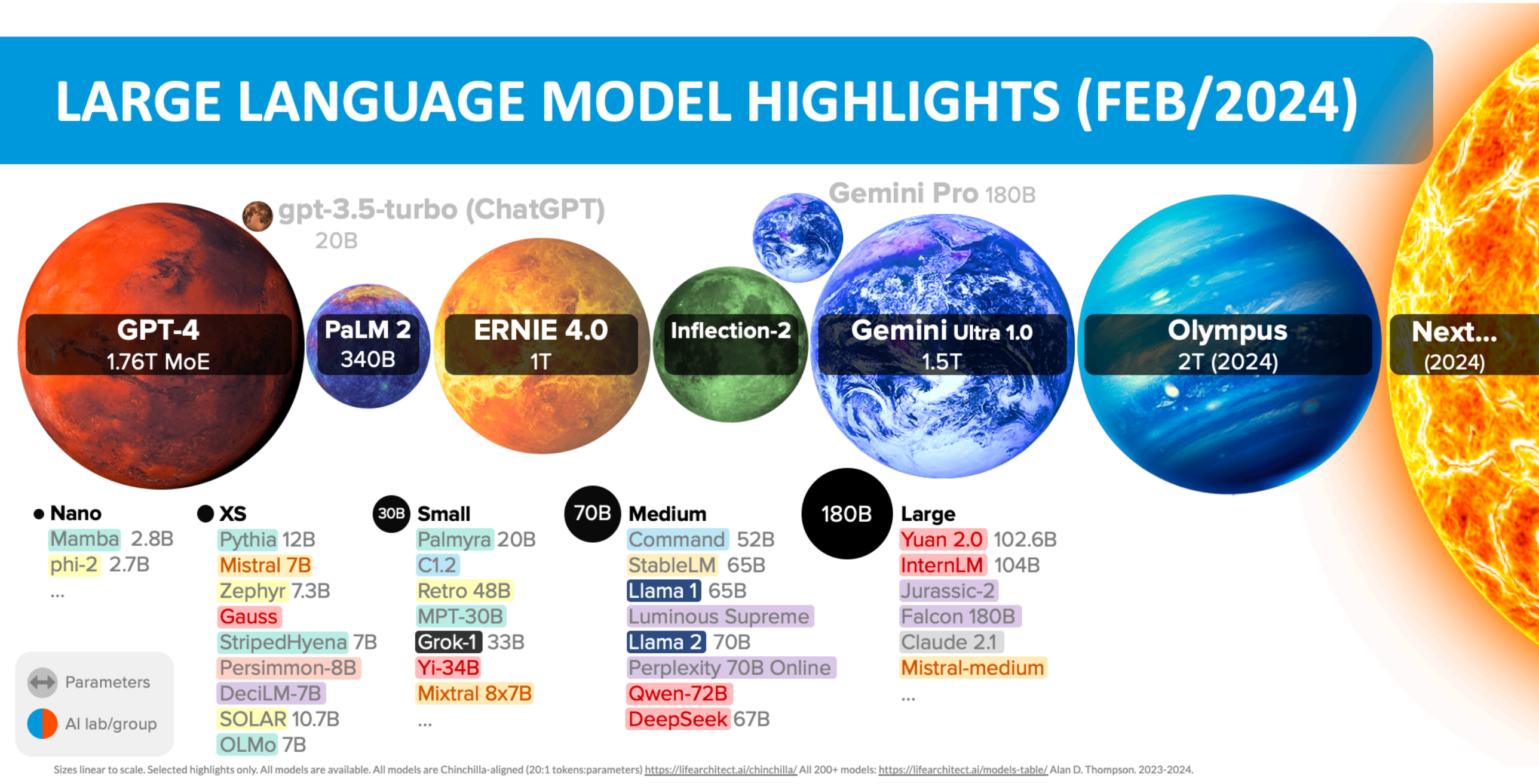
- LLMs are… LARGE!



Large Language Models

https://pureinsights.com/blog/2023/what-are-large-language-models-llms-search-and-ai-perspectives/

- LLMs are… LARGE!



Large Language Models

https://pureinsights.com/blog/2023/what-are-large-language-models-llms-search-and-ai-perspectives/

- LLMs are… LARGE! And getting larger…

Transformers

- More than meets the eye!

Transformers

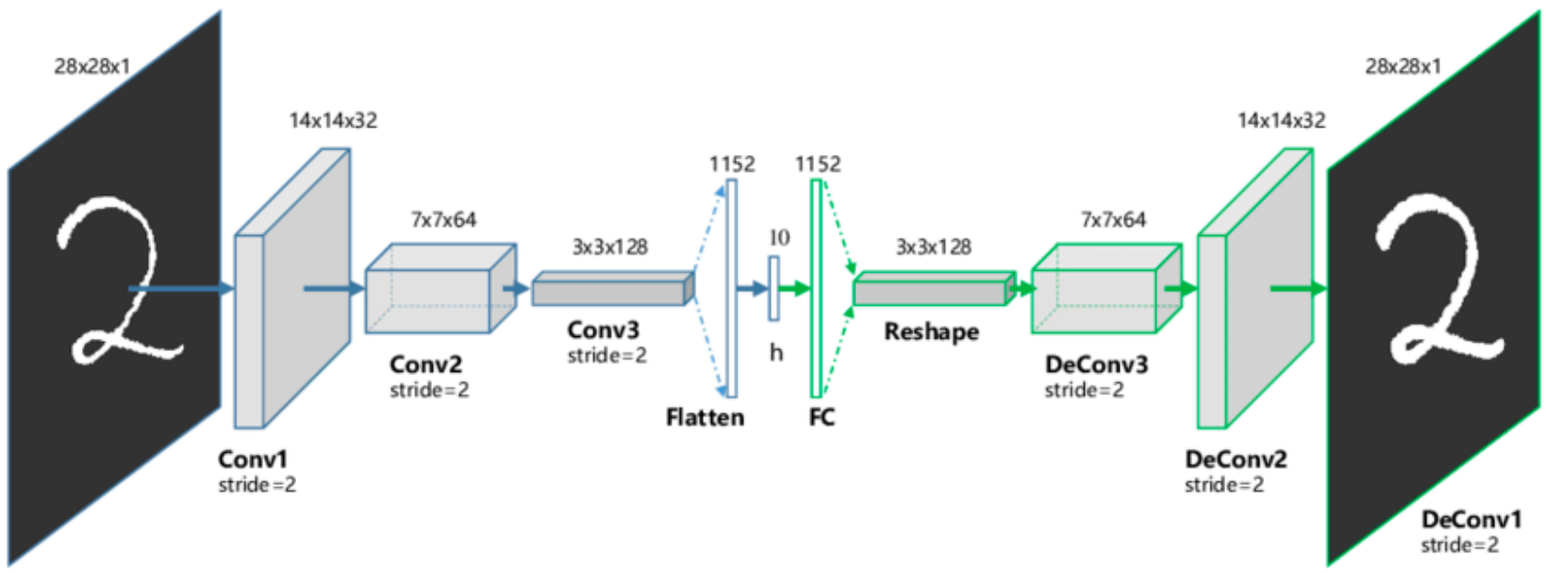
arXiv:1706.03762 (2017)

- More than meets the eye!
- "Attention is all you need" (2017), by A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin
- Foundational model for current state-of-the-art models in NLP, CV, Reinforcement Learning, Speech, etc…
- Overcomes limitations of previous SOTA models, like Recurrent Neural Networks and Convolutional Neural Networks
- Uses Encoder/Decoder architecture with Attention
- mechanisms to capture context.

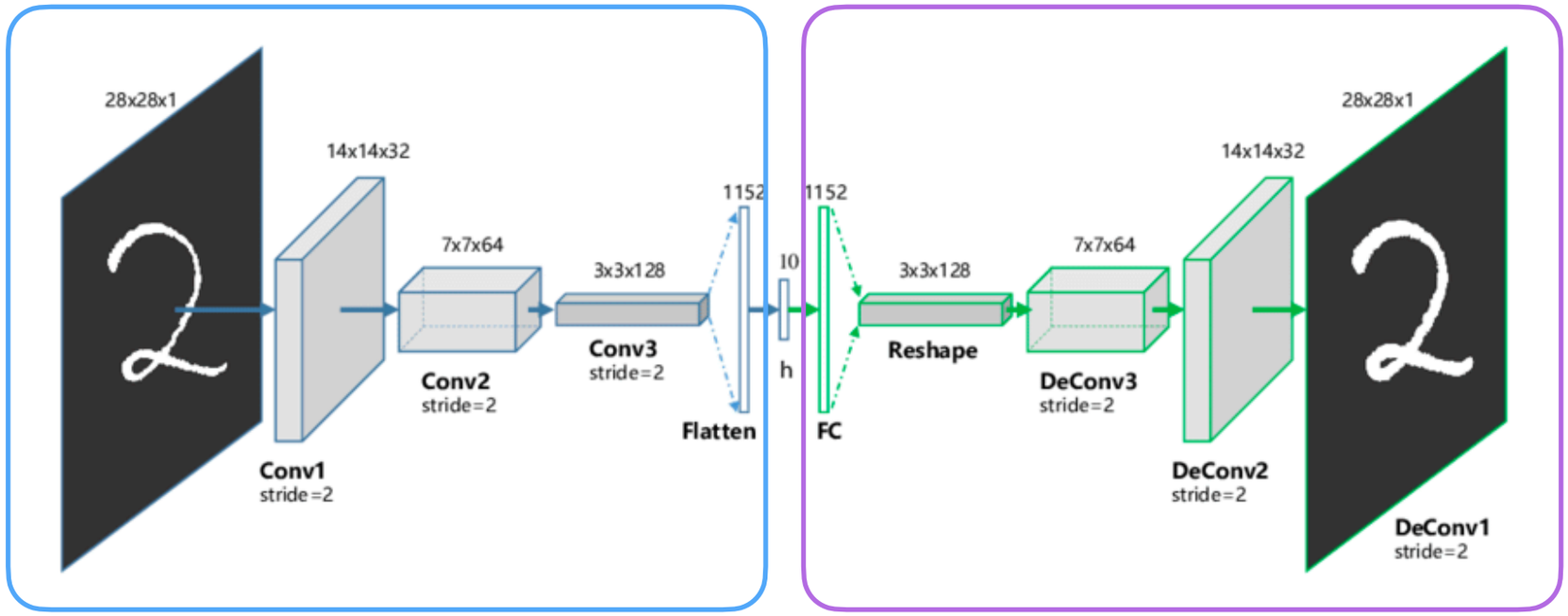
Auto-Encoders

https://www.researchgate.net/figure/The-structure-of-proposed-Convolutional-AutoEncoders-CAE-for-MNIST-In-the-middle-there_fig1_320658590

- After training, the parts of the network that generate the internal representation can be used as inputs to other Networks
- This is similar to what we did when we reused the word embeddings generated by training a word2vec network
- Auto-encoders can be arbitrarily complex, including many layers between the input and the internal representation (or Code), and are often used in Image Processing to generate efficient representations of complex images



Encoder/Decoder Architecture



Encoder

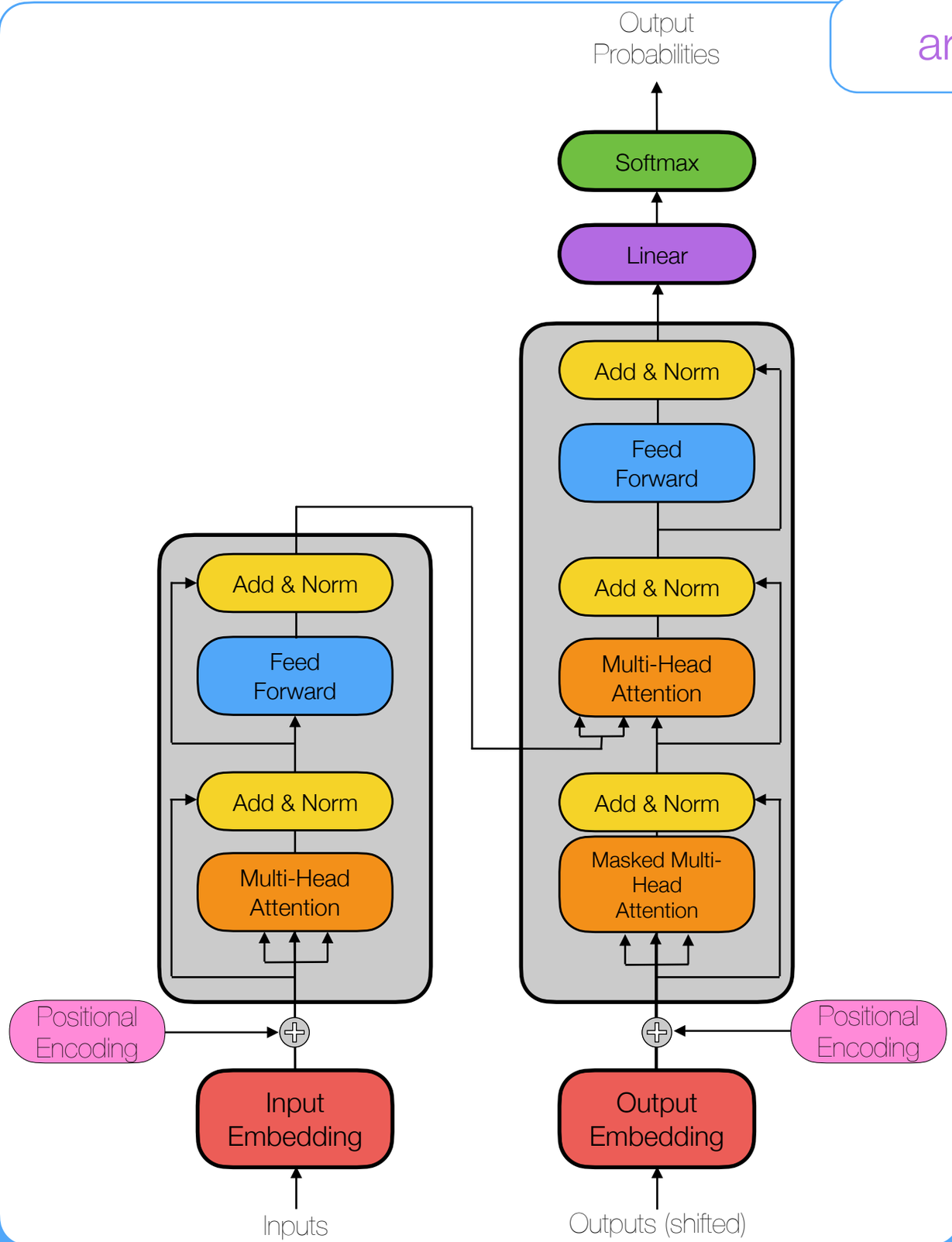
Input -> Internal Representation

Decoder

Internal Representation -> Output

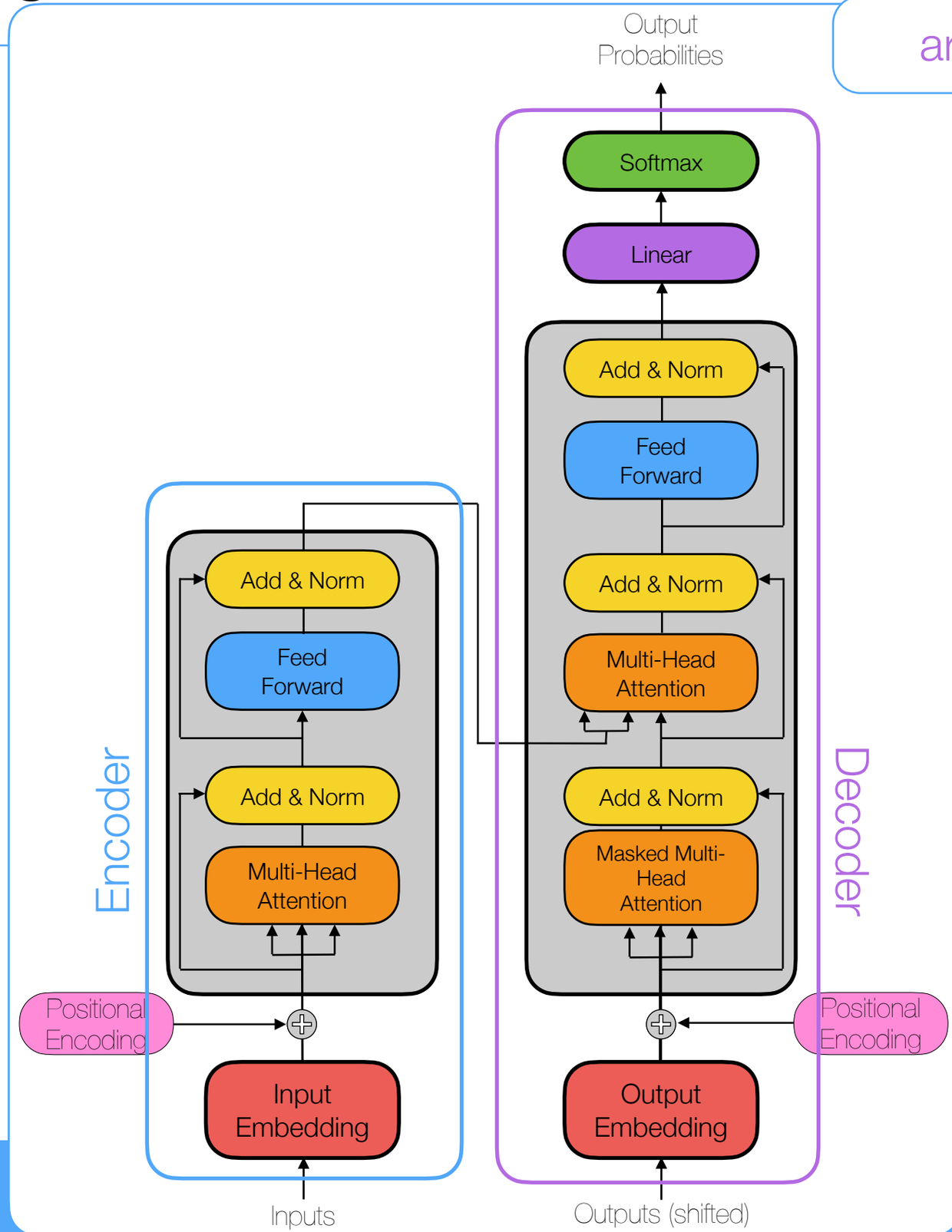
Transformers

arXiv:1706.03762 (2017)



Transformers

- Builds on the Encoder/Decoder architecture

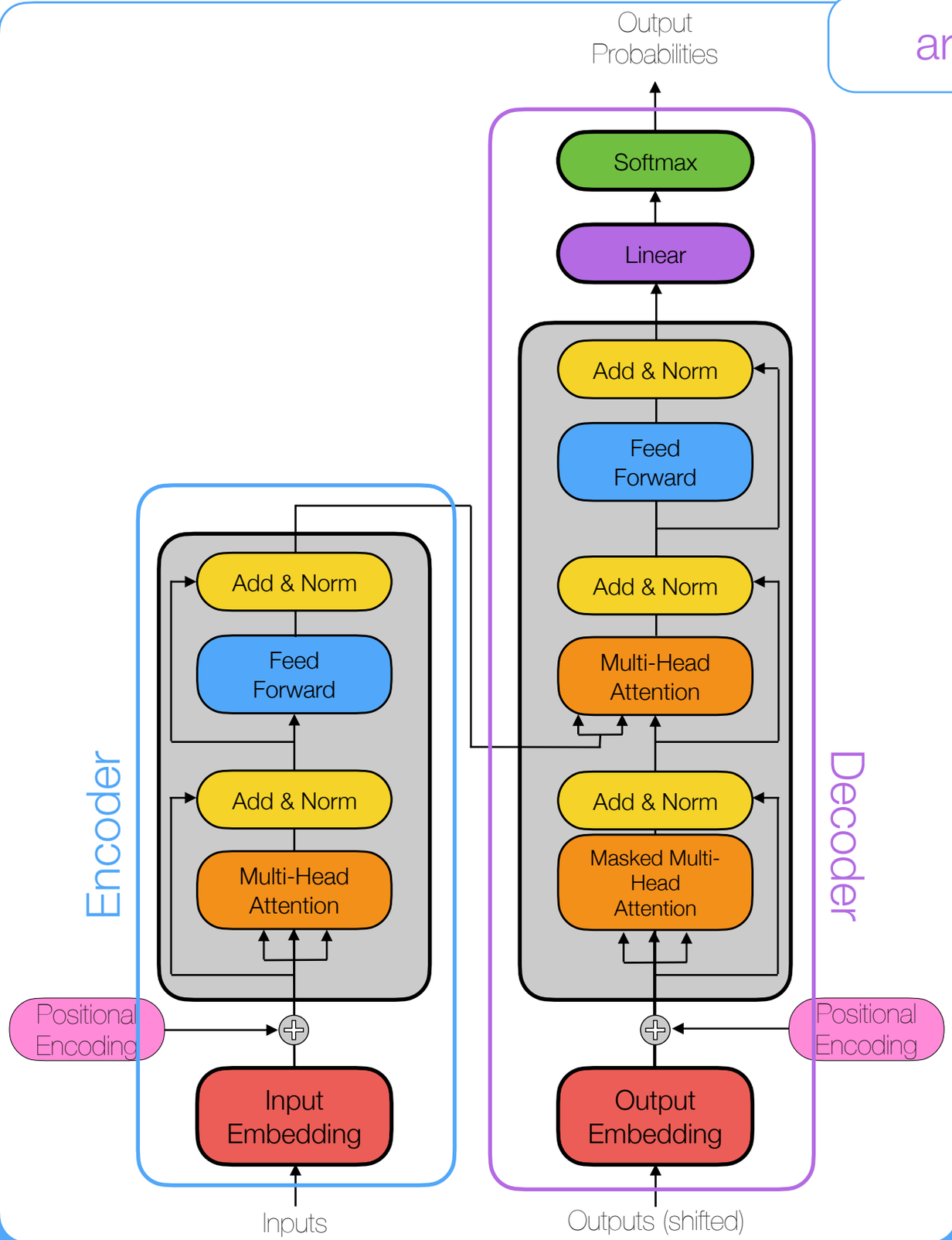


arXiv:1706.03762 (2017)

Transformers

- Builds on the Encoder/Decoder architecture
- Relies on the Attention mechanism

Multi-Head Attention



arXiv:1706.03762 (2017)

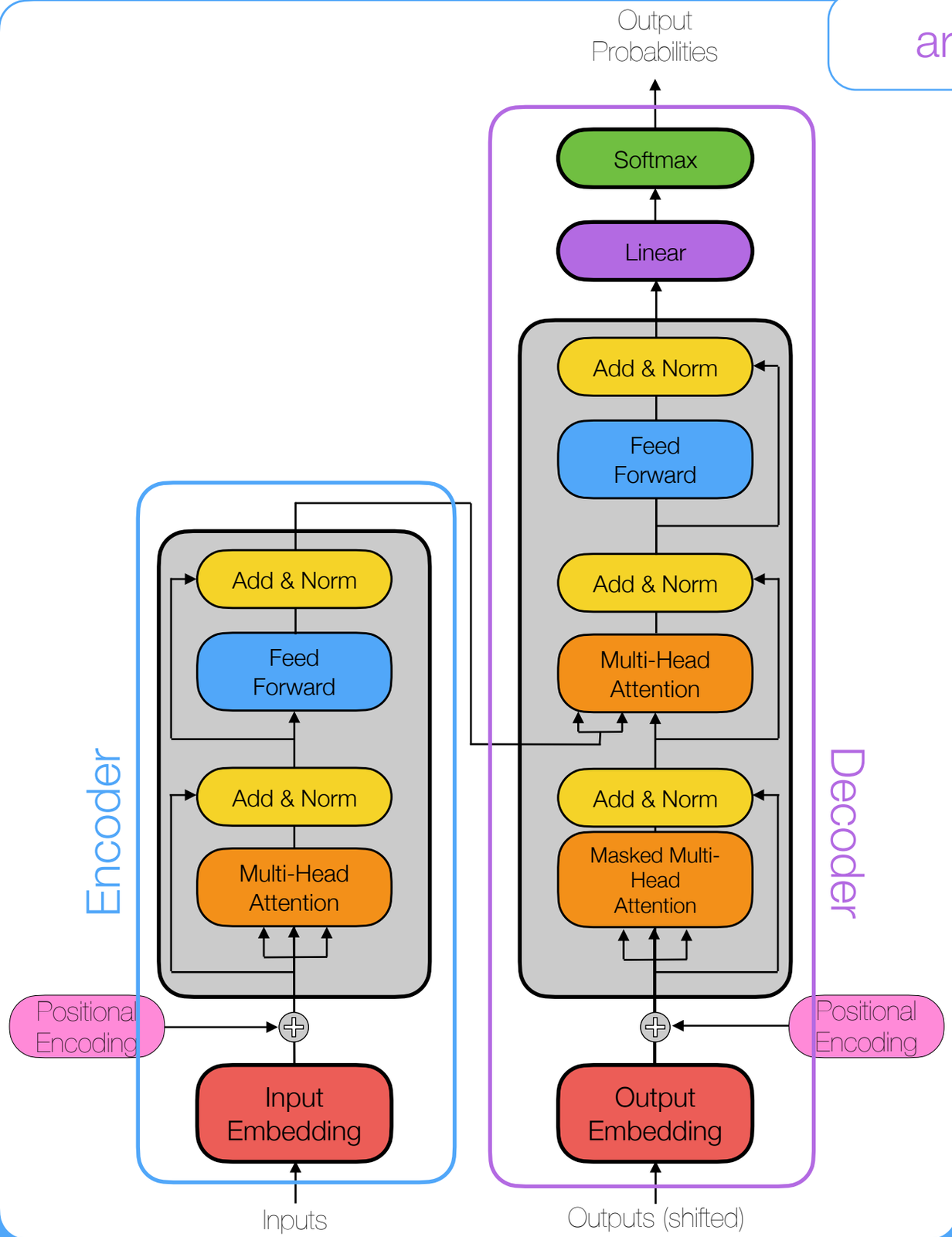
Transformers

- Builds on the Encoder/Decoder architecture
- Relies on the Attention mechanism

Multi-Head Attention

- And Embeddings with Relative Position Information

Positional Encoding



arXiv:1706.03762 (2017)

Different Kinds of Language Modeling

- Language models are trained on a sequence of tokens and asked to guess tokens have have been masked

- There are three ways to do this:

BERT Architecture

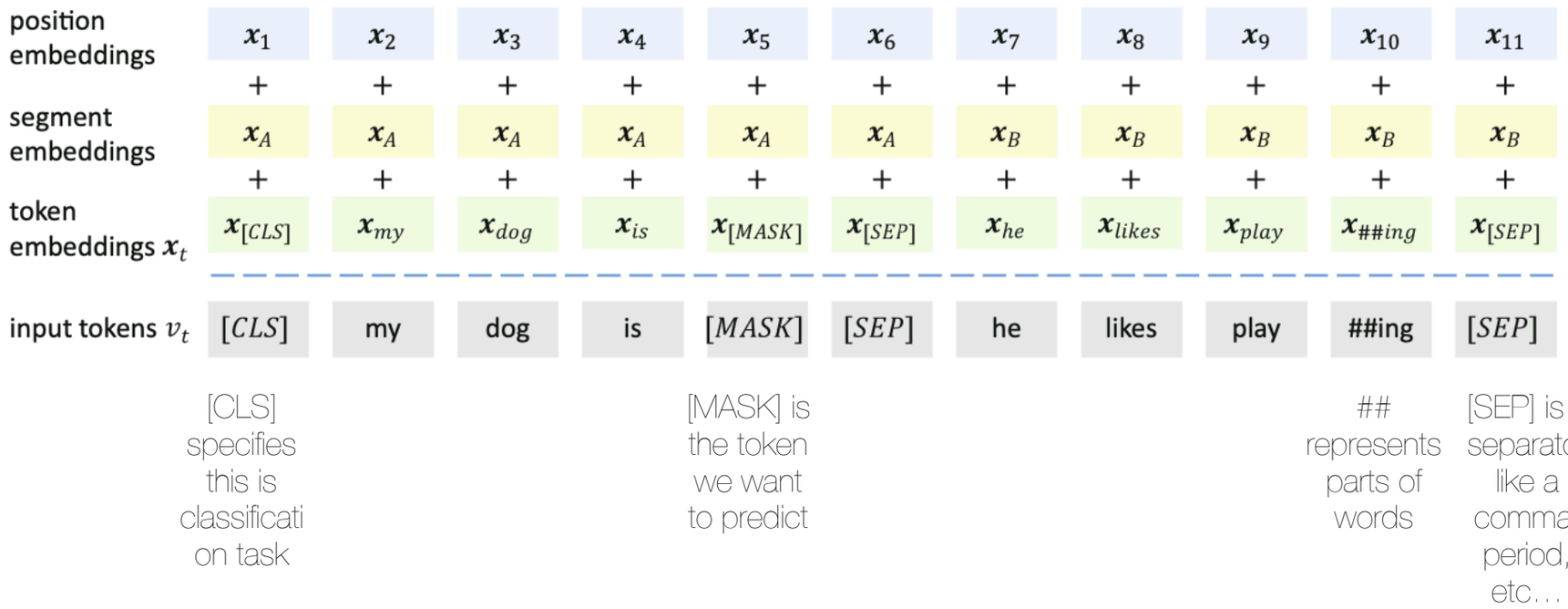
arXiv.1810.04805 (2018)

- BERT
  - Bidirectional - considers context from both left and right directions
  - Encoder - Includes only the Encoder component
  - Representations from
  - Transformers
- Designed for natural language understanding (NLU) tasks.
- Trained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP)
  - MLM - the model is trained to predict random tokens in the input that have been masked based on the context provided by the surrounding tokens
  - NSP - the model is trained to predict whether a given sentence follows another sentence in a pair.
- Has multiple layers of transformers (12 for BERT-base and 24 for BERT-large)
- Opensource!

BERT Architecture

Positional Encoding

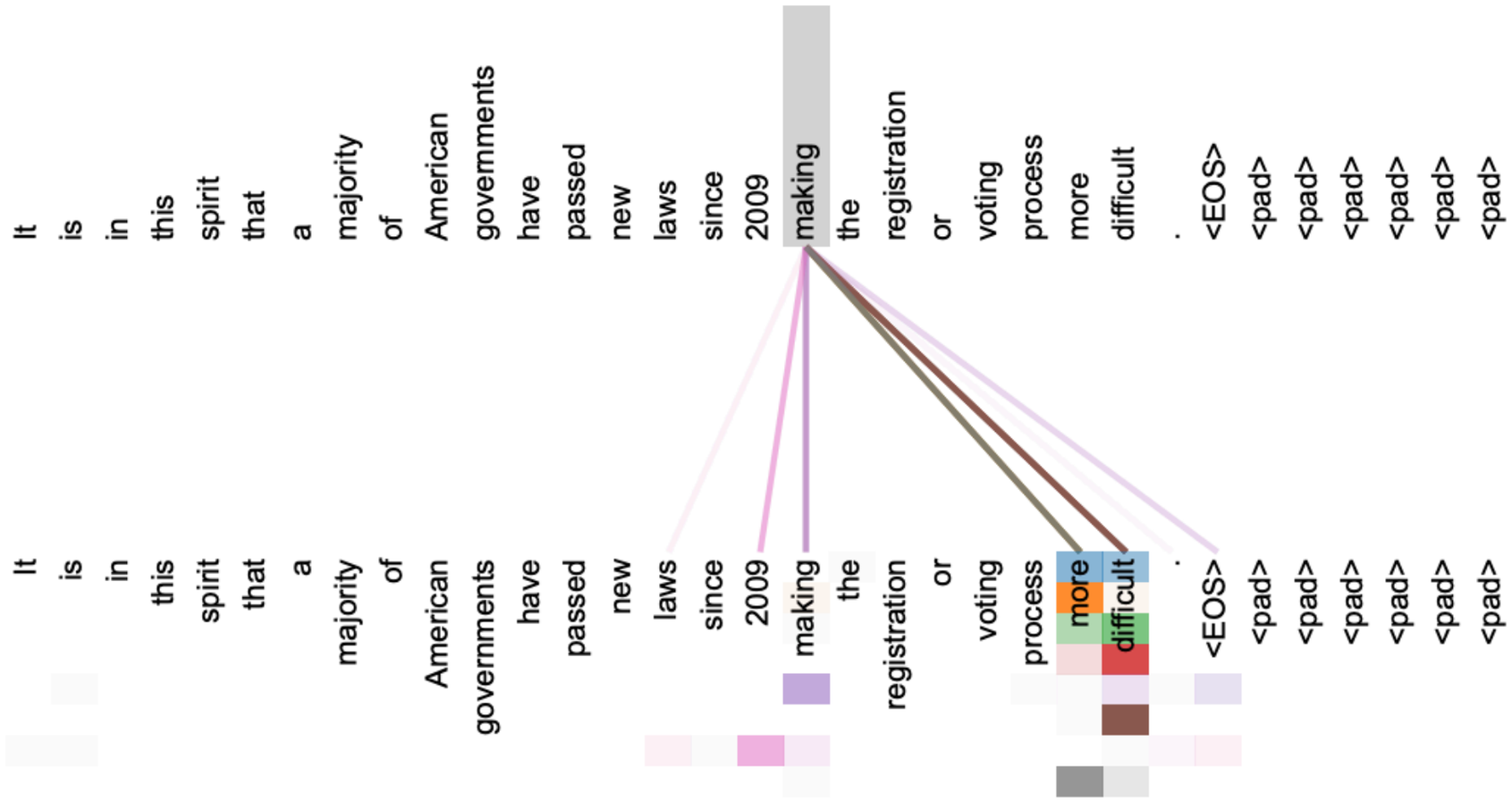
- Input tokens are fed into the model as embeddings after being modified to include position and segment information



Attention

arXiv:1706.03762 (2017)

- "Simple" mechanism to allow each token to take the context it appears in into account
- Requires exponentially more weights to be computed (from each token to every other token)



- Weights indicate the importance of each word relative to the current one

BertViz

https://github.com/DataForScience/ChatGPT

Attention

arXiv:1706.03762 (2017)

- The attention mechanism relies on 3 vectors, $\boldsymbol{k}$, $\boldsymbol{q}$ and $\boldsymbol{v}$. If $\boldsymbol{x}$ is the input token embedding

  - Key Vectors ($\boldsymbol{k}$) - representations of the input sequence that are used to compute the importance or relevance of each element in the input sequence

$$k_t^T = k_t^T W^{(k)}$$

  - Query Vectors ($\boldsymbol{q}$) - representations of the target sequence (or the same input sequence in the case of self-attention) that are used to determine which parts of the input sequence are relevant to each position in the target sequence.

$$q_t^T = x_t^T W^{(q)}$$

  - Value Vectors ($\boldsymbol{v}$) - representations of the input sequence that contain the actual information to be attended to or utilized in the output

$$v_t^T = k_t^T W^{(v)}$$

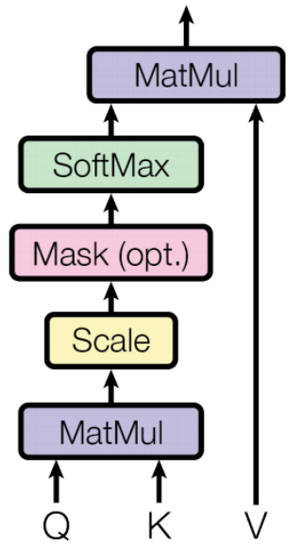
Attention

arXiv:1706.03762 (2017)

- The attention score for each token is then given by $\frac{q_r^T k_i}{\sqrt{d_k}}$

- These scores are then normalized to a probability $\alpha_{r,i}$

$$(\alpha_{r,1}, \ldots, \alpha_{r,T}) = \text{softmax}\left(\frac{\boldsymbol{q}_r^\intercal \boldsymbol{k}_1}{\sqrt{d_k}}, \ldots, \frac{\boldsymbol{q}_r^\intercal \boldsymbol{k}_T}{\sqrt{d_k}}\right)$$



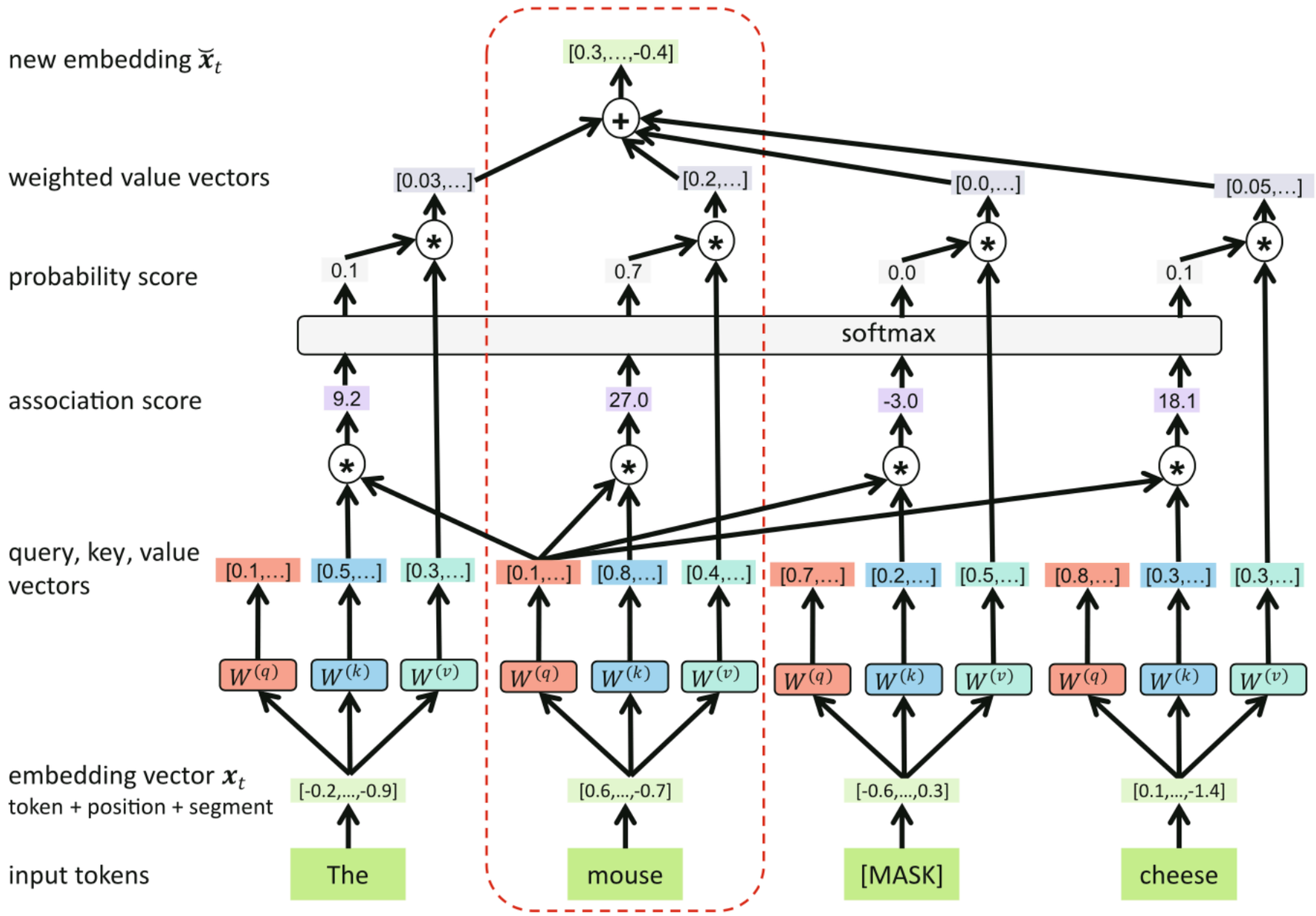
- That is finally used to compute the contextual embeddings as a weighted average of the value vectors, $\boldsymbol{v}$, and probability values $\boldsymbol{\alpha}$

$$\breve{\boldsymbol{x}}_r = \alpha_{r,1} * \boldsymbol{v}_1 + \cdots + \alpha_{r,T} * \boldsymbol{v}_T$$

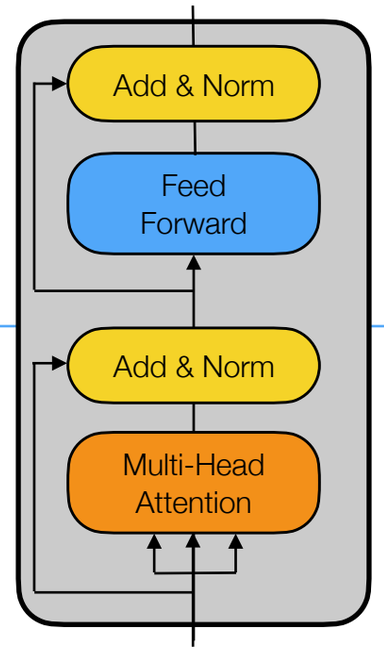
Contextual Embeddings

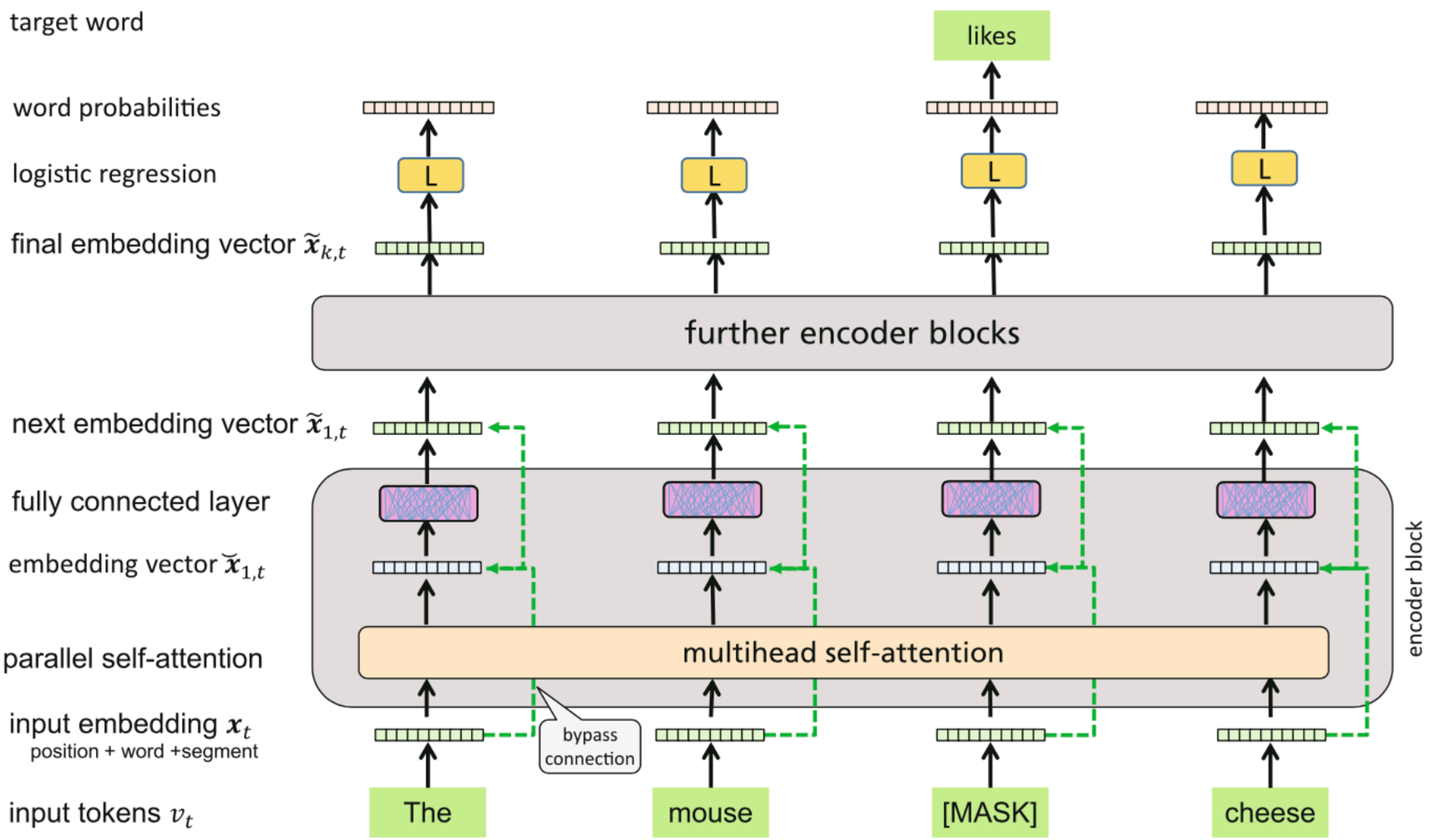
Multi-Head Attention

- The attention mechanism is then used to generate contextual embeddings that take



BERT Architecture



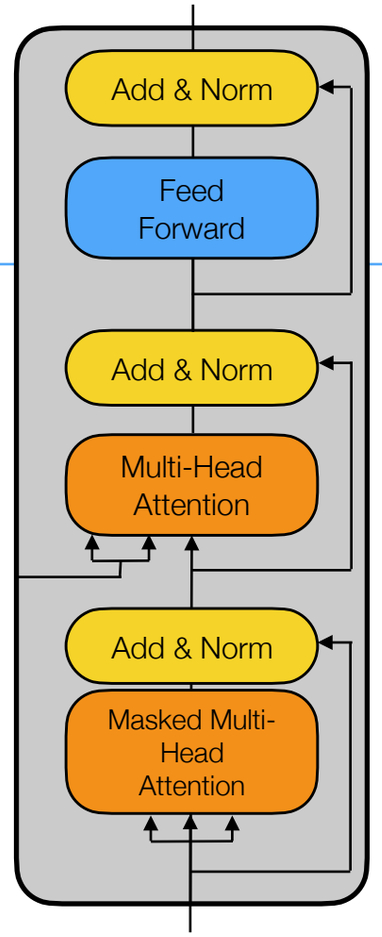


Bert + Transformers

https://github.com/DataForScience/ChatGPT

GPT Architecture

- GPT
  - Generative - Optimized for generating conversational responses
  - Pre-Trained - Trained on a large Corpus
  - Transformer - Only the Decoder portion used for generation
- Unidirectional - Processes text sequentially, from left to right.
- Masked Self Attention to prevent using unseen "future" tokens
- Can be fine-tuned for specific tasks by further training on task-specific data, adjusting parameters, or using techniques like transfer learning
- Optimized for dialog



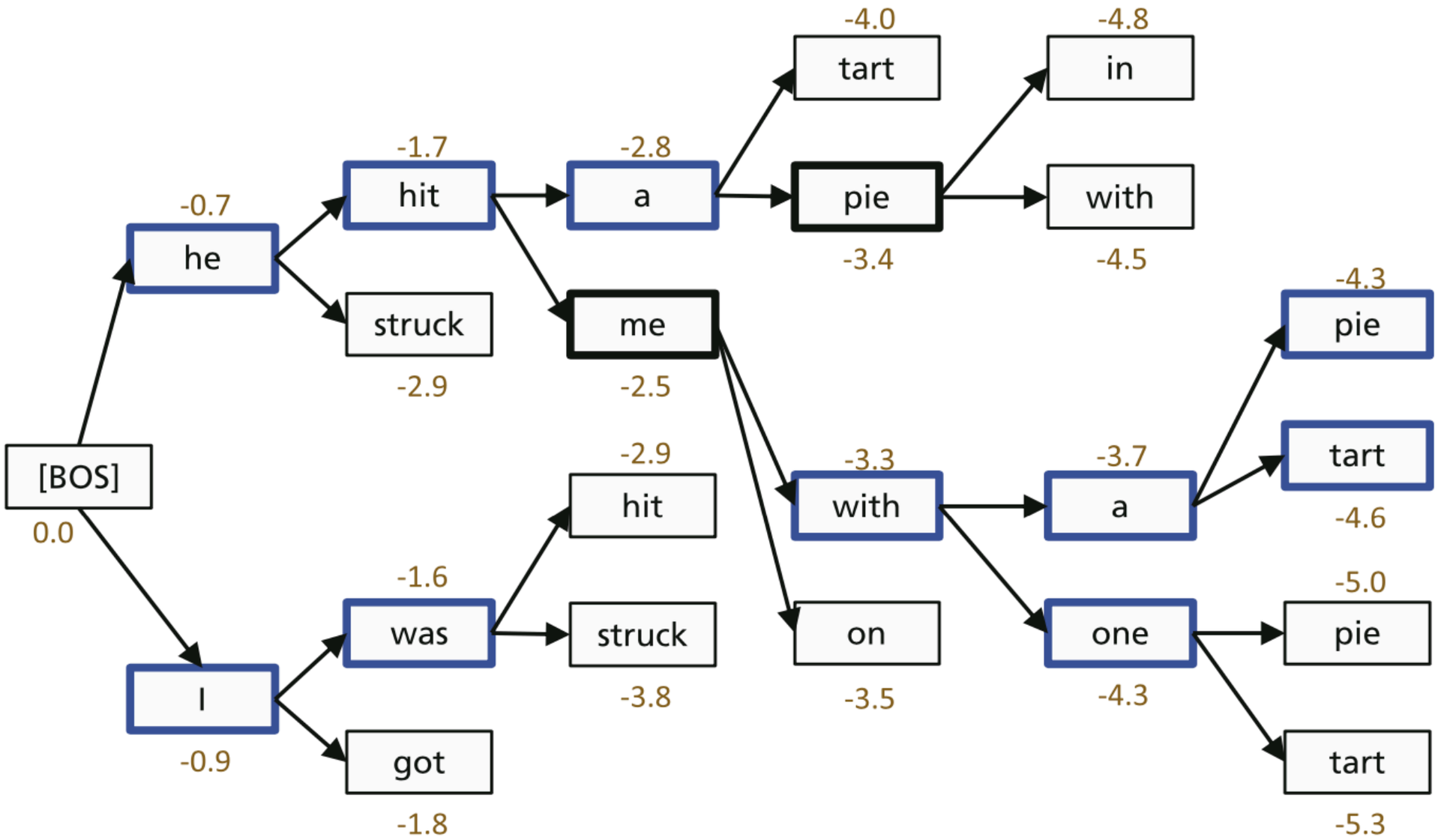
GPT Family

OpenAI's "GPT-n" series

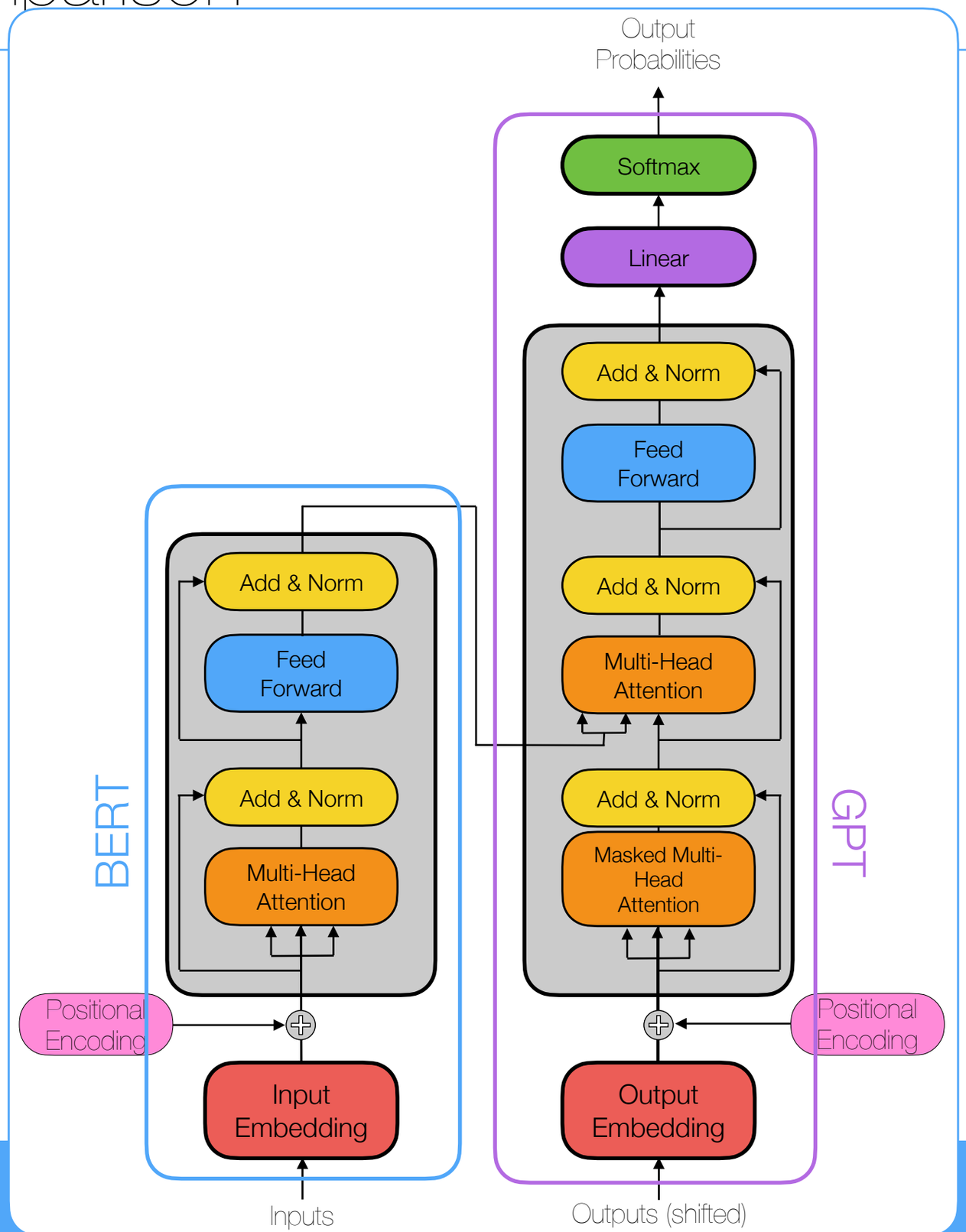
Model	Architecture	Parameter count	Training data	Release date	Training cost
GPT-1	12-level, 12-headed Transformer decoder (no encoder), followed by linear-softmax.	117 million	BookCorpus:[34] 4.5 GB of text, from 7000 unpublished books of various genres.	June 11, 2018[9]	30 days on 8 P600 GPUs, or 1 petaFLOP/s-day.[9]
GPT-2	GPT-1, but with modified normalization	1.5 billion	WebText: 40 GB of text, 8 million documents, from 45 million webpages upvoted on Reddit.	February 14, 2019 (initial/limited version) and November 5, 2019 (full version)[35]	"tens of petaflop/s-day",[36] or 1.5e21 FLOP.[37]
GPT-3	GPT-2, but with modification to allow larger scaling	175 billion[38]	499 billion tokens consisting of CommonCrawl (570 GB), WebText, English Wikipedia, and two books corpora (Books1 and Books2).	May 28, 2020[36]	3640 petaflop/s-day (Table D.1[36]), or 3.1e23 FLOP.[37]
GPT-3.5	Undisclosed	175 billion[38]	Undisclosed	March 15, 2022	Undisclosed
GPT-4	Also trained with both text prediction and RLHF; accepts both text and images as input. Further details are not public.[33]	Undisclosed. Estimated 1.7 trillion[39]	Undisclosed	March 14, 2023	Undisclosed. Estimated 2.1e25 FLOP.[37]

Text Generation

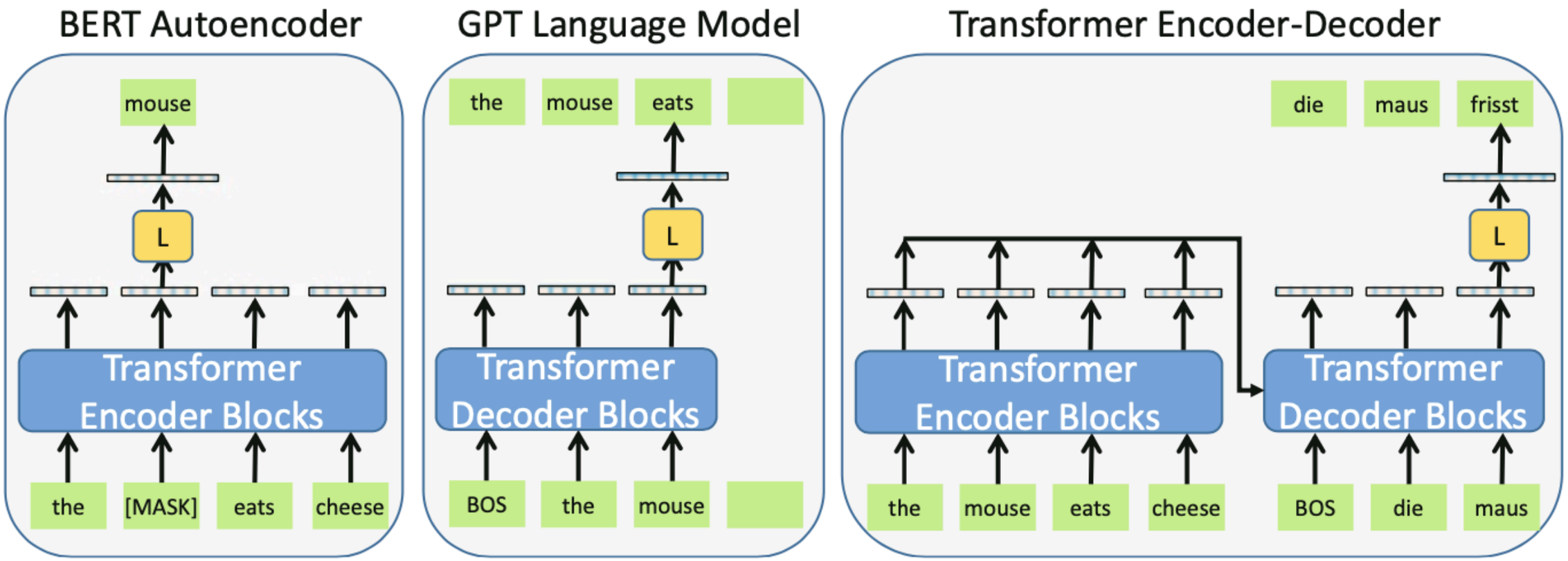
- A random walk in token space



Model Comparison



Model Comparison



Model Comparison

Aspect	BERT	GPT-3.5
Architecture	Bidirectional Transformer Encoder	Autoregressive Transformer Decoder
Training Data	Pre-trained on large corpora, fine-tuned for specific tasks	Pre-trained on large corpora, not fine-tuned for tasks
Fine-tuning	Effective for various NLP tasks with fine-tuning	Not explicitly designed for fine-tuning tasks
Context	Contextual understanding at word-level	Contextual understanding at word and sequence level
Generation	Not designed for text generation tasks	State-of-the-art in text generation tasks
Comprehension	Good at understanding semantics and relationships	Exceptional at understanding and generating coherent text
Task-specific	Requires task-specific fine-tuning for optimal performance	General-purpose model, less need for task-specific fine-tuning
Embeddings	Provides rich word embeddings for downstream tasks	Doesn't provide word embeddings but generates text
Scalability	Less scalable due to bidirectional nature	Highly scalable due to autoregressive decoding
Memory efficiency	Requires less memory during inference	Requires more memory due to autoregressive nature
Training time	Faster training time compared to GPT models	Longer training time due to autoregressive decoding
Flexibility	Limited flexibility in generating text	Highly flexible in generating diverse text outputs

Large Language Models

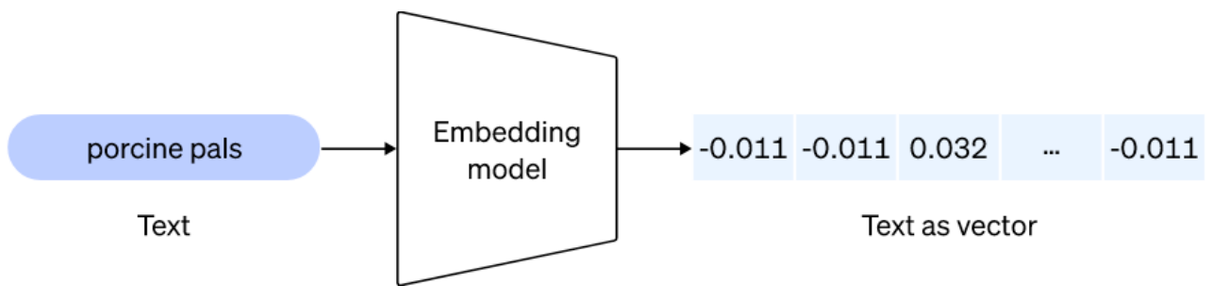
https://github.com/DataForScience/ChatGPT

3. Embeddings

Understanding Embeddings

https://openai.com/blog/new-and-improved-embedding-model

- Embeddings are a fundamental concept in natural language processing (NLP)
- An embedding is simply a mapping between a piece of text (a word, a sentence, etc) and a dense numerical vector

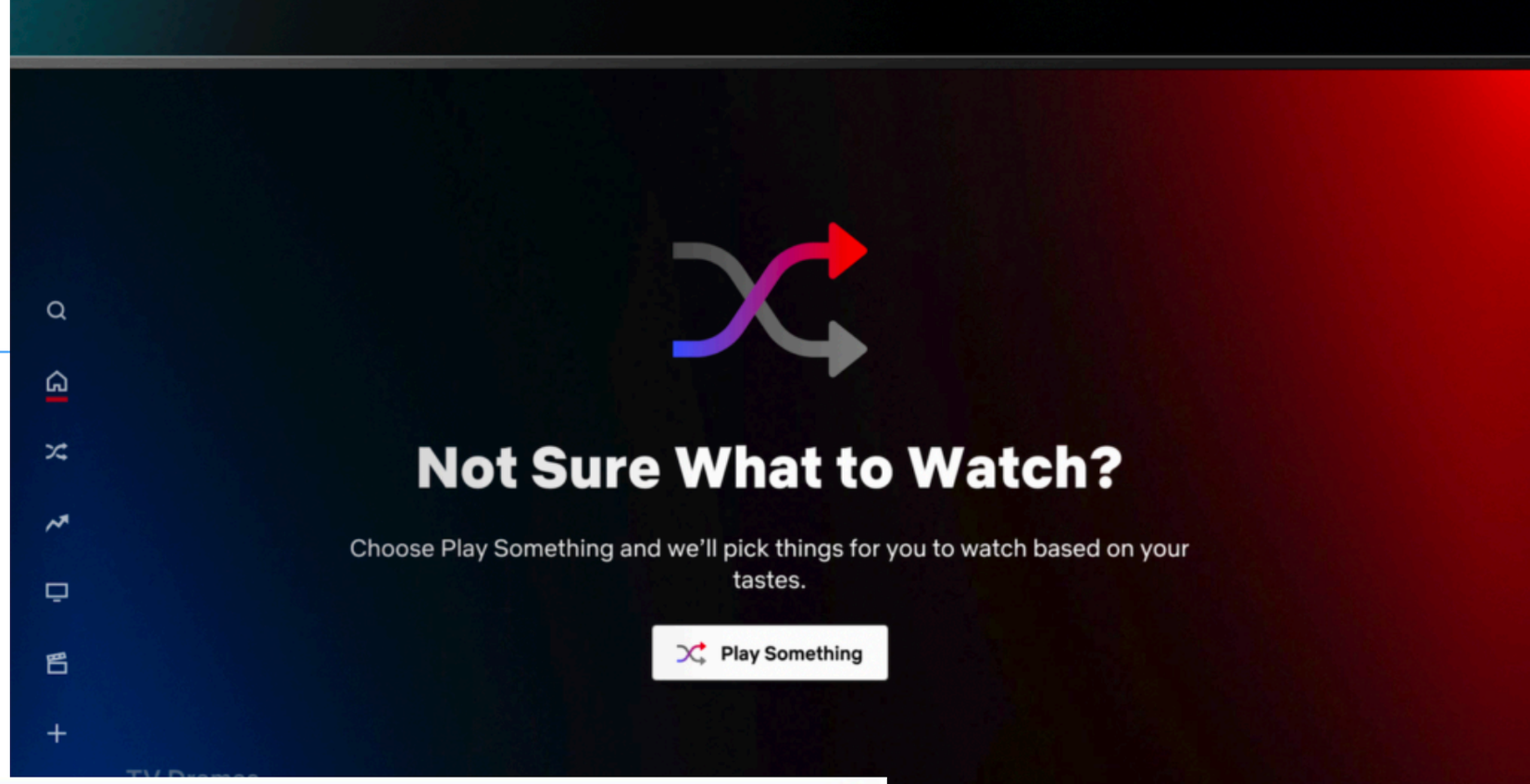


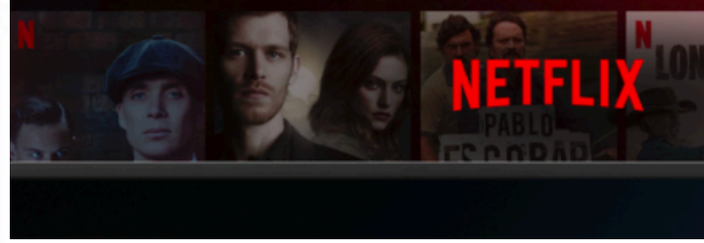
- Typical vector representations are;
  - one-hot encoding
  - bag of words
  - TF/IDF
  - etc
- None of these representations include semantic information

Understanding Embeddings

- Different techniques were developed to generate vector representations that explicitly encode semantics and that can be reused. Common ones are:
  - word2vec - Developed by Google using a simple Neural Network architecture.
  - GloVe - Developed by Stanford to explicitly take co-occurrences into account
  - Transformers
- Each vector encodes information about the meaning of the word it's associated with
- Embedding models work at the token level and are trained on a specific encoding of the tokens generated by a given tokenizer, so we must be careful to use the correct encoding
- Similarities between vectors match well to similarities between words
- There are several distance metrics we can use to compute the similarity between two vectors. The most common one is Cosine Similarity that measures the cosine of the angle between two vectors

Recommendations





amazon.com

Recommended for You

Amazon.com has new recommendations for you based on items you purchased or told us you own.

Google Apps Deciphered: Compute in the Cloud to Streamline Your Desktop

Google Apps Administrator Guide: A Private-Label Web Workspace

Googlepedia: The Ultimate Google Resource (3rd Edition)

Embeddings

https://github.com/DataForScience/ChatGPT

4. Applications Outside NLP

CODEX Model

https://openai.com/blog/openai-codex

- OpenAI released the CODEX model in Aug 2021.
- It was a version of GPT-3 that was trained specifically in text and source code from GitHub.
- Designed to power the functionality of GitHub Copilot:https://github.com/features/copilot a digital assistant for programmers. In particular, CODEX was capable of
    - Produce code based on a prompt
    - Autocomplete your code as your writing it
    - Suggest a useful library or API call for an application
    - Comment pre-existing code
    - Improve the efficiency of existing code
- In Mar 2023, CODEX was folded in to the general GPT-3.5-turbo model so all of this functionality is now available within the system we're already familiar with.

DALL-E

arXiv:2204.06125 (2022)

- DALL-E takes textual descriptions of scenes, objects, concepts, or even abstract ideas as input.

- Embeddings are used to generate a high-dimensional vector representation that captures semantic information from the text and provides the model with the context it needs to generate relevant images.

- The model generates images pixel by pixel by mapping the textual embeddings to corresponding image features.

- DALL-E is trained on a large dataset of text-image pairs by minimizing a loss function that measures the similarity between the generated images and the ground truth images in the dataset.

- Attention Mechanisms allow it to focus on specific words or phrases in the input text and generate corresponding image details accordingly.

DALL-E

arXiv:2204.06125 (2022)

- Image models like DALL-E build on top of the LLM approach outlined above
- Essentially, DALL-E replaces the LLM decoder with an image decoder
- The image decoder is trained using Contrastive Training:
  - A large list of images and textual descriptions is used to train the image decoder to generate an internal representation that matches the encoded description

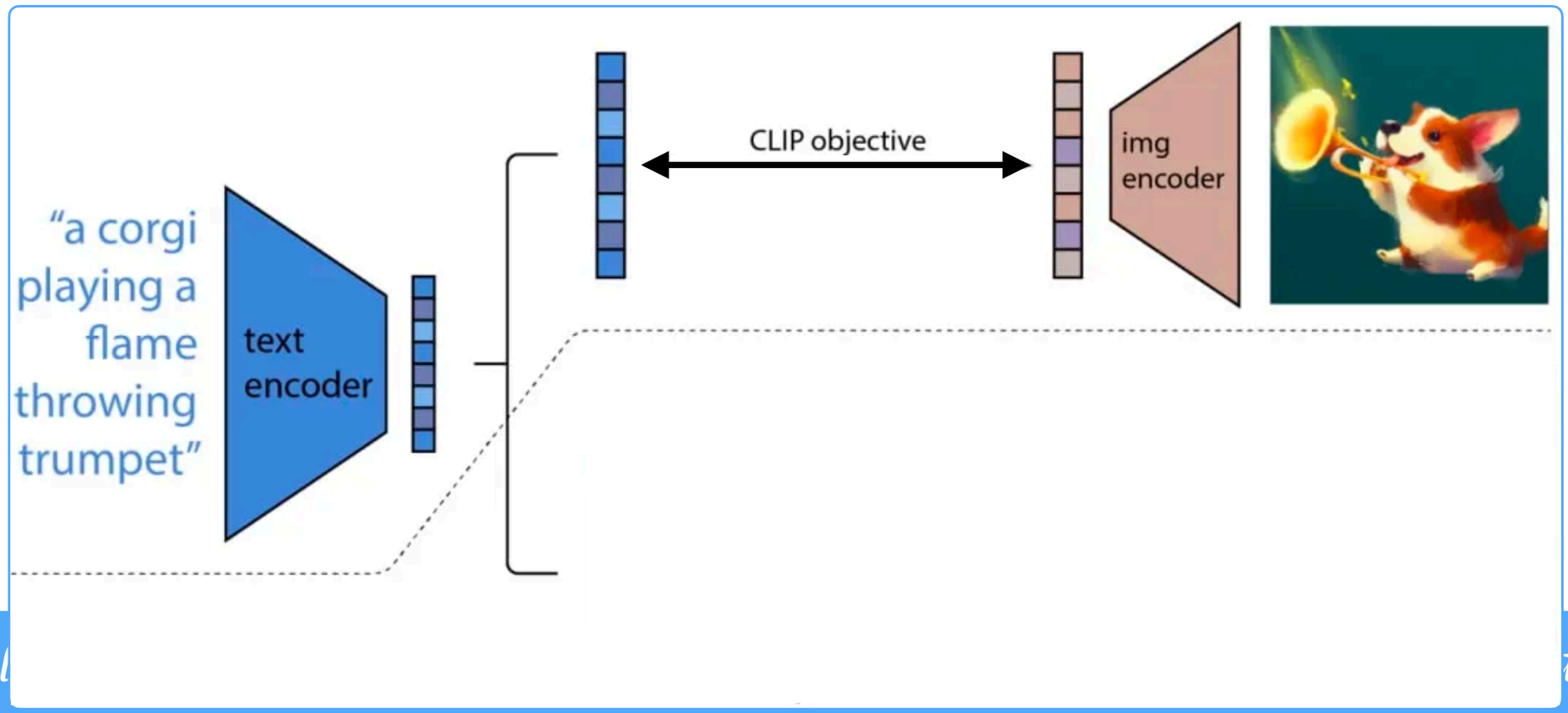


Image Models

arXiv:2204.06125

- Image models like DALL-E build on top of the LLM approach outlined above
- Essentially, DALL-E replaces the LLM decoder with an image decoder
- The image decoder is trained using Contrastive Training:
  - A large list of images and textual descriptions is used to train the image decoder to generate an internal representation that matches the encoded description
  - Inverting the process produces a model that converts textual descriptions into images

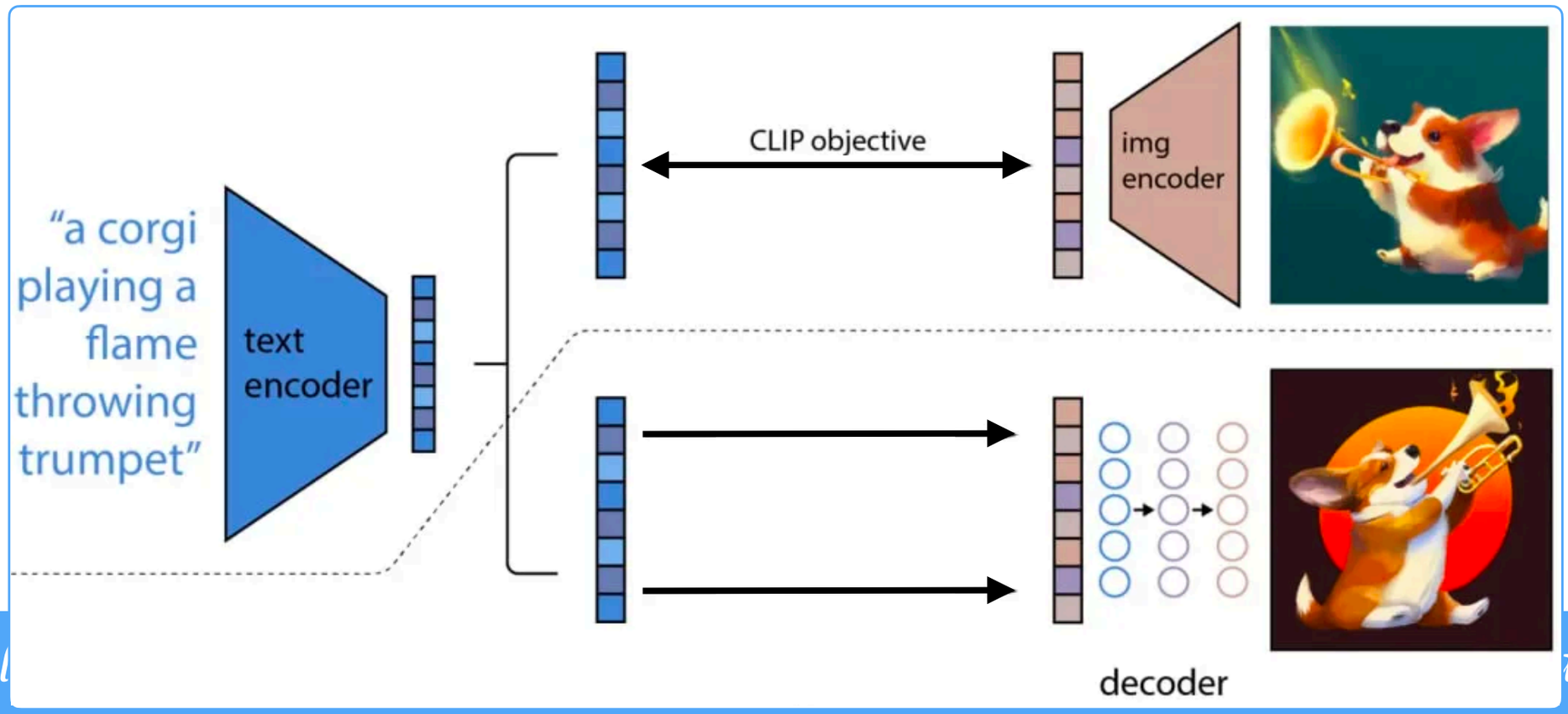
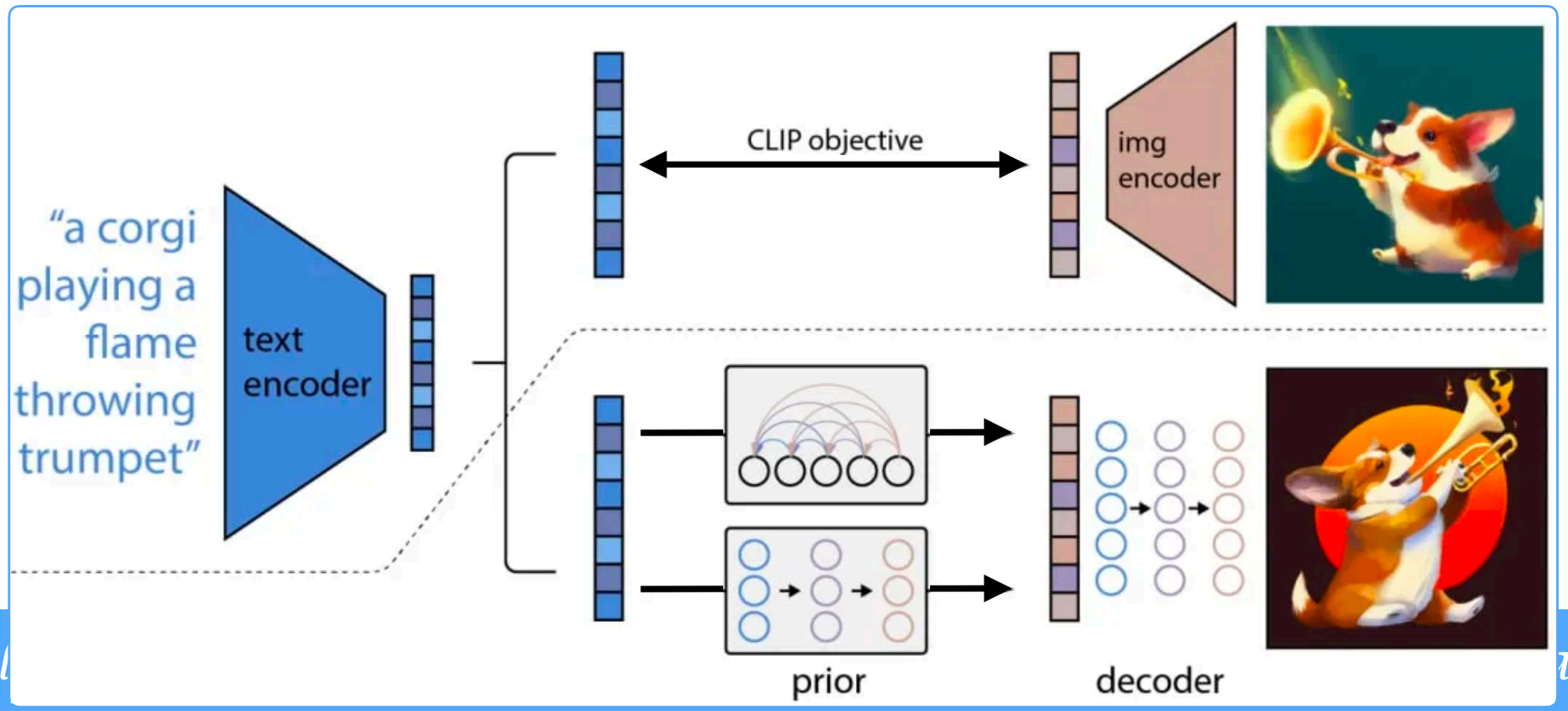


Image Models

arXiv:2204.06125 (2022)

- Image models like DALL-E build on top of the LLM approach outlined above
- Essentially, DALL-E replaces the LLM decoder with an image decoder
- The image decoder is trained using Contrastive Training:
  - A large list of images and textual descriptions is used to train the image decoder to generate an internal representation that matches the encoded description
  - Inverting the process produces a model that converts textual descriptions into images
- As a final improvement, we add a prior in between the output of the text encoder and the input of the image decoder.



BloombergGPT

arXiv:2303.17564 (2023)

- An LLM with 50 billion parameters specifically trained on a wide range of financial data.

Date	Bloomberg	Filings	News	Press	Web	*Total*
2007 [03-]	276	73	892	523	2,667	*4,431*
2008	351	91	1,621	628	9,003	*11,695*
2009	293	93	1,791	528	9,179	*11,883*
2010	292	111	1,917	527	11,388	*14,236*
2011	335	117	2,264	548	13,643	*16,907*
2012	403	105	2,502	529	15,015	*18,554*
2013	415	87	2,437	441	17,230	*20,610*
2014	396	251	2,458	437	18,510	*22,052*
2015	358	1,639	2,371	427	20,782	*25,576*
2016	324	1,891	2,509	418	24,337	*29,478*
2017	294	2,294	2,567	398	25,283	*30,837*
2018	275	1,791	2,702	420	26,027	*31,214*
2019	263	1,662	3,102	504	27,195	*32,726*
2020	277	1,632	2,794	805	30,928	*36,435*
2021	247	1,767	3,515	938	29,749	*36,215*
2022 [-07]	140	882	2,206	531	16,872	*20,631*
	4,939	*14,486*	*37,647*	*8,602*	*297,807*	*363,482*

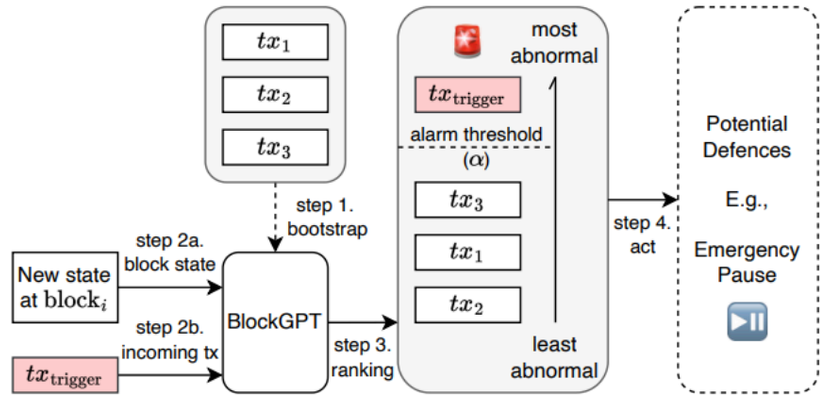
Table 2: The number of tokens (in millions) contained within documents in FINPILE, organized by year (rows) and type (column). Units are millions of tokens.

- It's capable of:
  - Generating Bloomberg Query Language (BQL)
  - Providing suggestions for news headlines
  - Answering financial questions
- Outperforms other LLMs on financial NLP tasks

BlockGPT

arXiv:2304.12749 (2003)

- GPT Variant trained on Ethereum Blockchain data
- Designed as a real-time approach to detecting anomalous blockchain transactions
- Was able to detect anomalous transactions for dozens of DeFi protocol attacks



Victim Name	Victim Contract	Application Categories	Damage (in USD)
Beanstalk	0xc1e0..24c5	Stablecoin	181,500,000
MonoX	0x66e7..ee63	DEX	31,133,333
PopsicleFinance	0xd63b..3546	Yield farming	20,700,000
PrimitiveFinance	0x9dae..f2f9	Derivatives	13,000,000
PunkProtocol	0x929c..49d6	Others	8,950,000
VisorFinance	0xc9f2..14ef	Others	8,200,000
DAOMaker	0xd6c8..b1ec	Others	4,000,000
DAOMaker	0x933f..2a13	Others	4,000,000
DODO	0x051e..a2b6	DEX	3,800,000
DODO	0x509e..41fb	DEX	3,800,000
CheeseBank	0x833e..743d	Digital Bank	3,300,000
dydx	0x5377..ba2c	Derivatives	2,211,000
RevestFinance	0xe952..1659	Others	2,005,000
BTFinance	0x3ec4..8af0	Yield farming	1,600,000
VisorFinance	0x65bc..054f	Others	975,720
WildCredit	0x7b3b..c6ca	Lending	650,000
SharedStake	0xa231..7ef5	Others	500,000
88mph	0x2165..b0a6	Lending	100,000
SanshuInu	0x35c6..7810	Others	100,000
KlondikeFinance	0xacbd..e747	Synthetic assets	22,116

Applications Outside NLP

https://github.com/DataForScience/ChatGPT

Question

- How was the technical level?
    - 1 — Too Low (too many details)
    - 2 — Low
    - 3 — Just Right
    - 4 — High
    - 5 — Too High (not enough details)

Question

- How was the level of Python code/explanations?
    - 1 — Too Low (too many details)
    - 2 — Low
    - 3 — Just Right
    - 4 — High
    - 5 — Too High (not enough details)

Events

graphs4sci.substack.com

NLP with Deep Learning for Everyone

Feb 28, 2023 - 10am-2pm (PST)

Interactive Data Visualization with Python

Mar 13, 2023 - 9am-1pm (PST)

NLP For Everyone

Apr 3, 2023 - 10am-2pm (PST)

VIDEO

Times Series Analysis for Everyone

★★★★★ 1 review

By Bruno Goncalves

https://bit.ly/Timeseries_LL

TIME TO COMPLETE:
6h

TOPICS:
Time Series

PUBLISHED BY:
Pearson

PUBLICATION DATE:
November 2021

Start

6 Hours of Video Instruction

The perfect introduction to time-based analytics

Overview

Times Series Analysis for Everyone LiveLessons covers the fundamental tools and techniques for the analysis of time series data. These lessons introduce you to the basic concepts, ideas, and algorithms necessary to develop your own time series applications in a step-by-step, intuitive fashion. The lessons follow a gradual progression, from the more specific to the more abstract, taking you from the very basics to some of the most recent and sophisticated algorithms by leveraging the statsmodels, arch, and Keras state-of-the-art models.

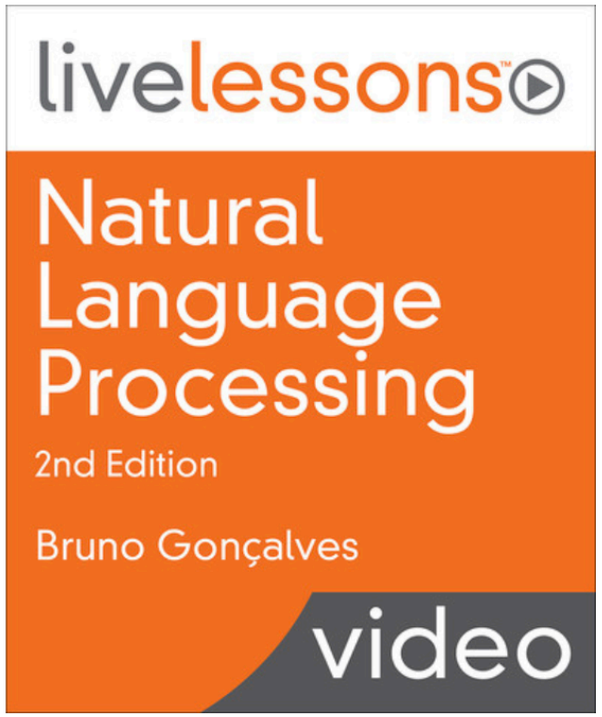
VIDEO

Natural Language Processing, 2nd Edition

Write the first review

By Bruno Goncalves

https://bit.ly/NLP_LL



TIME TO COMPLETE:
5h 23m

TOPICS:
Natural Language Processing

PUBLISHED BY:
Addison-Wesley Professional

PUBLICATION DATE:
October 2021

Start

5 Hours of Video Instruction

Overview

Natural Language Processing LiveLessons covers thefundamentals of Natural Language Processing in a simple and intuitive way,empowering you to add NLP to your toolkit. Using the powerful NLTK package, itgradually moves from the basics of text representation, cleaning, topicdetection, regular expressions, and sentiment analysis before moving on to theKeras deep learning framework to explore more advanced topics such as textclassification and sequence-to-sequence models. After successfully completingthese lessons you'll be equipped with a fundamental and practical understandingof state-of-the-art Natural Language Processing tools and algorithms.

END