LLM MECHANICS AND ARCHITECTURE GEN AI SERIES: PART 2

Sarah Rodenbeck

Lead Research Data Scientist

PURDUE UNIVERSITY | Rosen Center for Advanced Computing

GenAI Series

- Part 1: Prompt Engineering
  - Effectively using GenAI tools
  - Goal: Improve results by crafting better prompts
- Part 2: Underlying Mechanics and Architecture
  - Deep dive into how LLMs work
  - Goal: Provide a technical foundation to support critical evaluations of LLM advances
- Part 3: Fine tuning models
  - Creating custom chatbots
  - Goal: Explore tools and methods you can use to create a chatbot for a specific purpose or with custom data

Why learn the theory?

Foundation for further learning

Understanding why LLMs sometimes don’t work as expected

Context for evaluating how advances fit into the field

Quick Review of Key ML Terms

- **Unsupervised Learning**: neural network used to find patterns in unlabeled data, e.g., clustering
- **Supervised Learning**: Labelled data used to help the model "learn" how to do a particular task, e.g., classification
- **Transfer learning**: Reusing general information learned from a previous task for a new task; speeds up training and reduces data requirements
  - **Pre-training**: General learning
  - **Fine-tuning**: Tweaking the pre-trained model for a downstream task

Evolution of NLP

1950s-1990s

Symbolic NLP

- Expert rule-based systems hand-coded by linguists
- Key achievements:
  - Georgetown Experiment
  - ELIZA

1990s-2010

Statistical NLP

- Use similarities between words to compete tasks
- Key achievements:
  - Statistical Machine Translation
  - Latent Semantic Indexing/TFIDF
  - First use of NN for language modelling

2010s-Present

Neural NLP

- Rapid advancement in NLP thanks to more data and hardware
- Key achievements:
  - Word embeddings
  - Attention and Transformer
  - Large language models and GPT

Evolution of NLP

Pre-transformer

Post-transformer

Key Techniques/Models:

- Word2Vec
- SkipGram
- GloVe
- RNNs
- LTSMs

Key Techniques/Models:

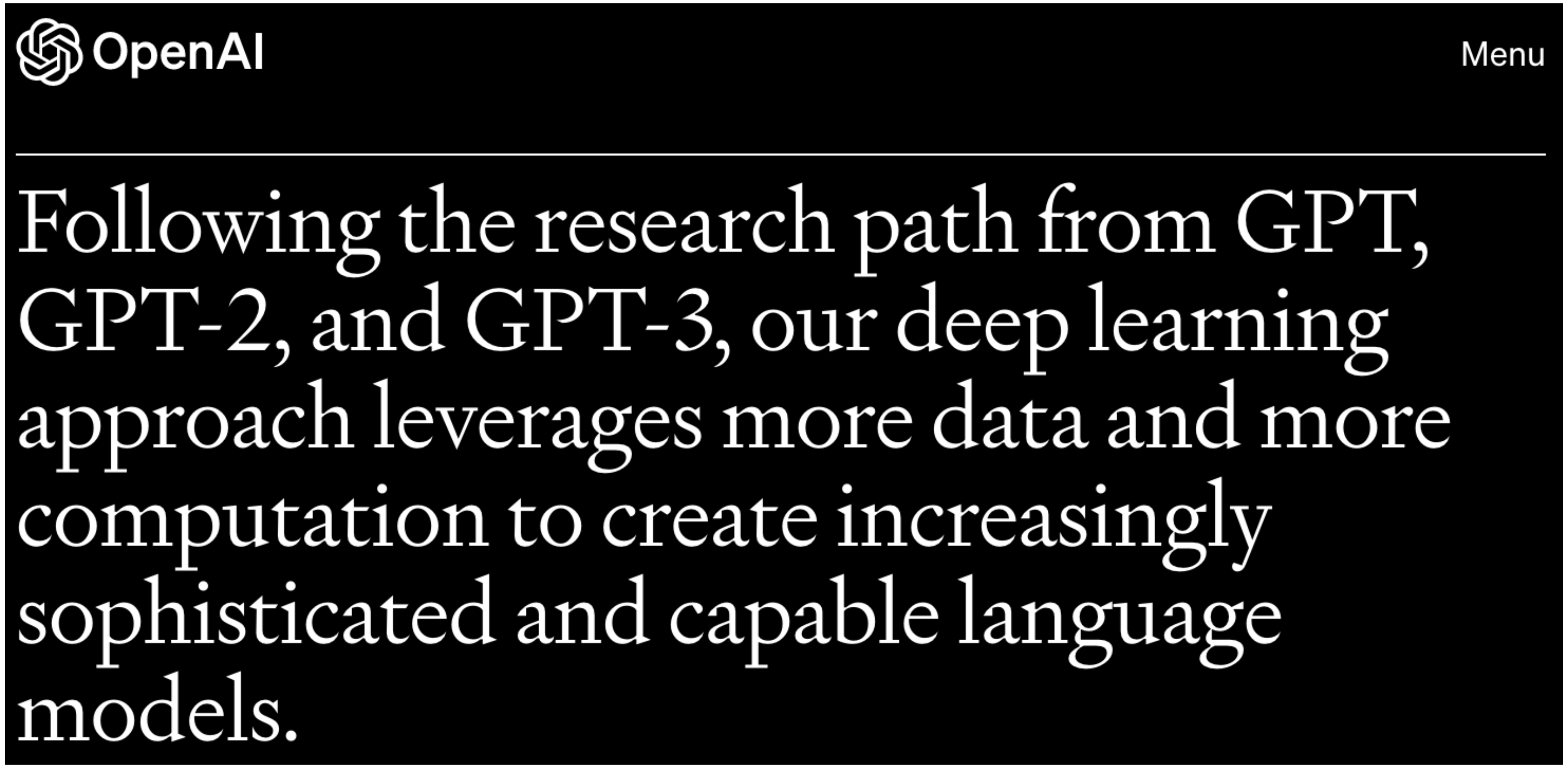
- Transformer
- Attention
- BERT
- GPT

Foundations

Key Idea of NLP

With a sufficiently large corpus of text data, models can learn the patterns of language

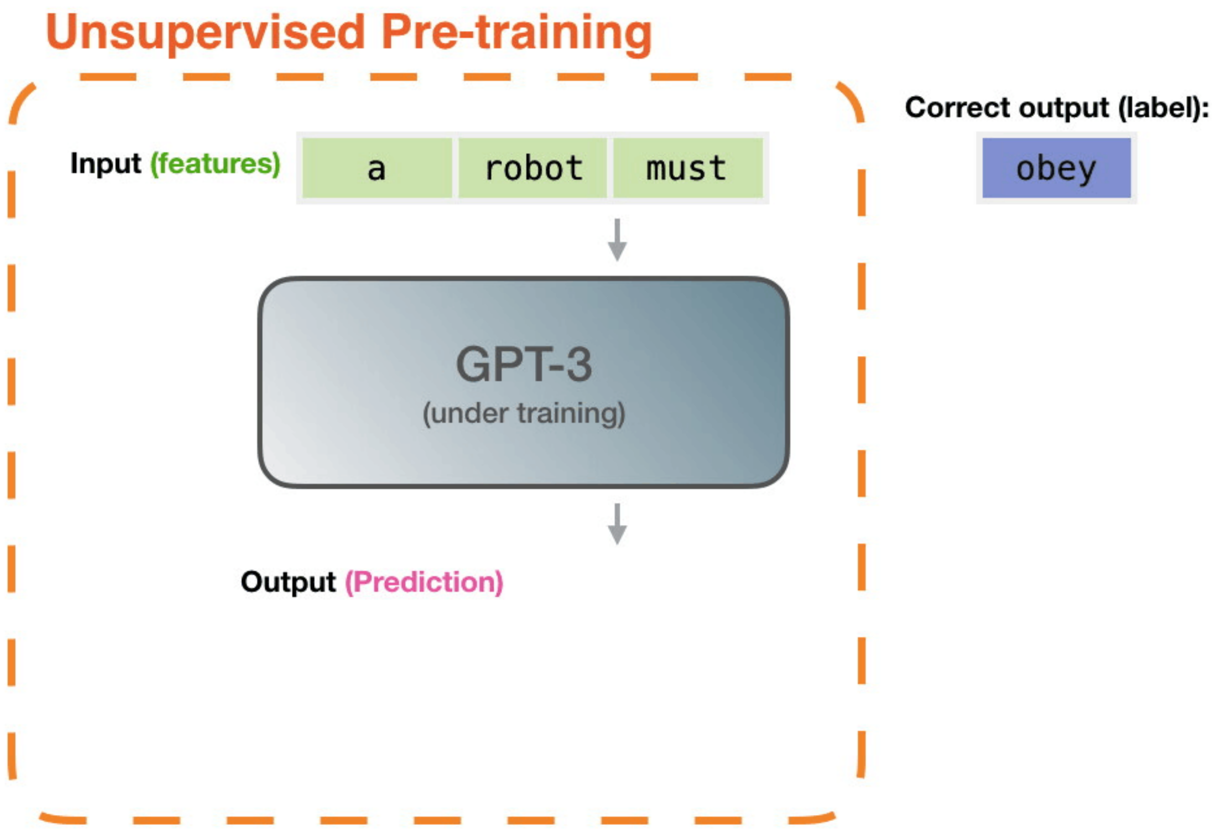
Data and Compute are the “Secret Sauce” for LLMs



Following the research path from GPT, GPT-2, and GPT-3, our deep learning approach leverages more data and more computation to create increasingly sophisticated and capable language models.

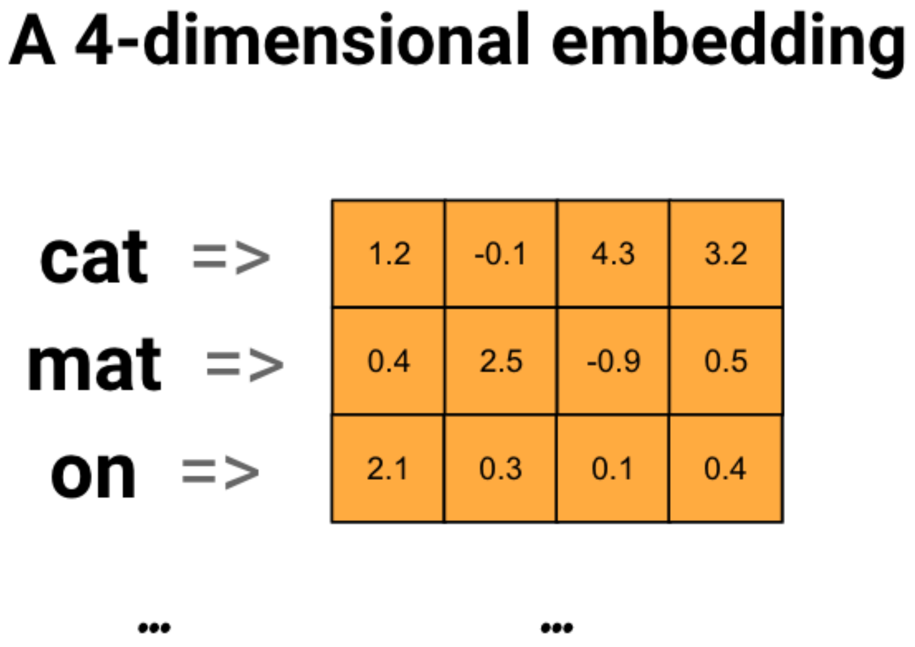
How do you Train a LLM?

- Start with random parameters
- Use unsupervised pre-training to find correct parameters
- Resulting model can be further fine-tuned for other specific tasks



Credit: Jay Alammar

Text as Numbers



A 4-dimensional embedding

cat	=>	1.2	-0.1	4.3	3.2
mat	=>	0.4	2.5	-0.9	0.5
on	=>	2.1	0.3	0.1	0.4
...		...			

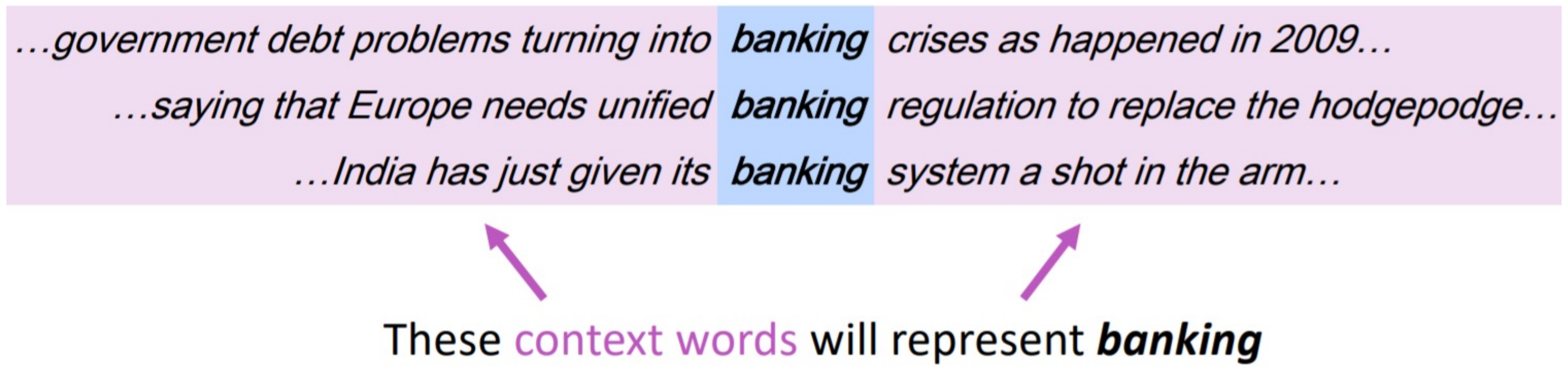
Embedding ideally captures:

- Meaning of words
- Similarities/differences between words
- Contextual meaning of words

Text as Numbers

“You shall know a word by the company it keeps”

- A word’s meaning can be understood based on the words it frequently appears close to
- Use the many contexts of a word to build up its representation


…government debt problems turning into ***banking*** *crises as happened in 2009…*
…saying that Europe needs unified ***banking*** *regulation to replace the hodgepodge…*
…India has just given its ***banking*** *system a shot in the arm…*

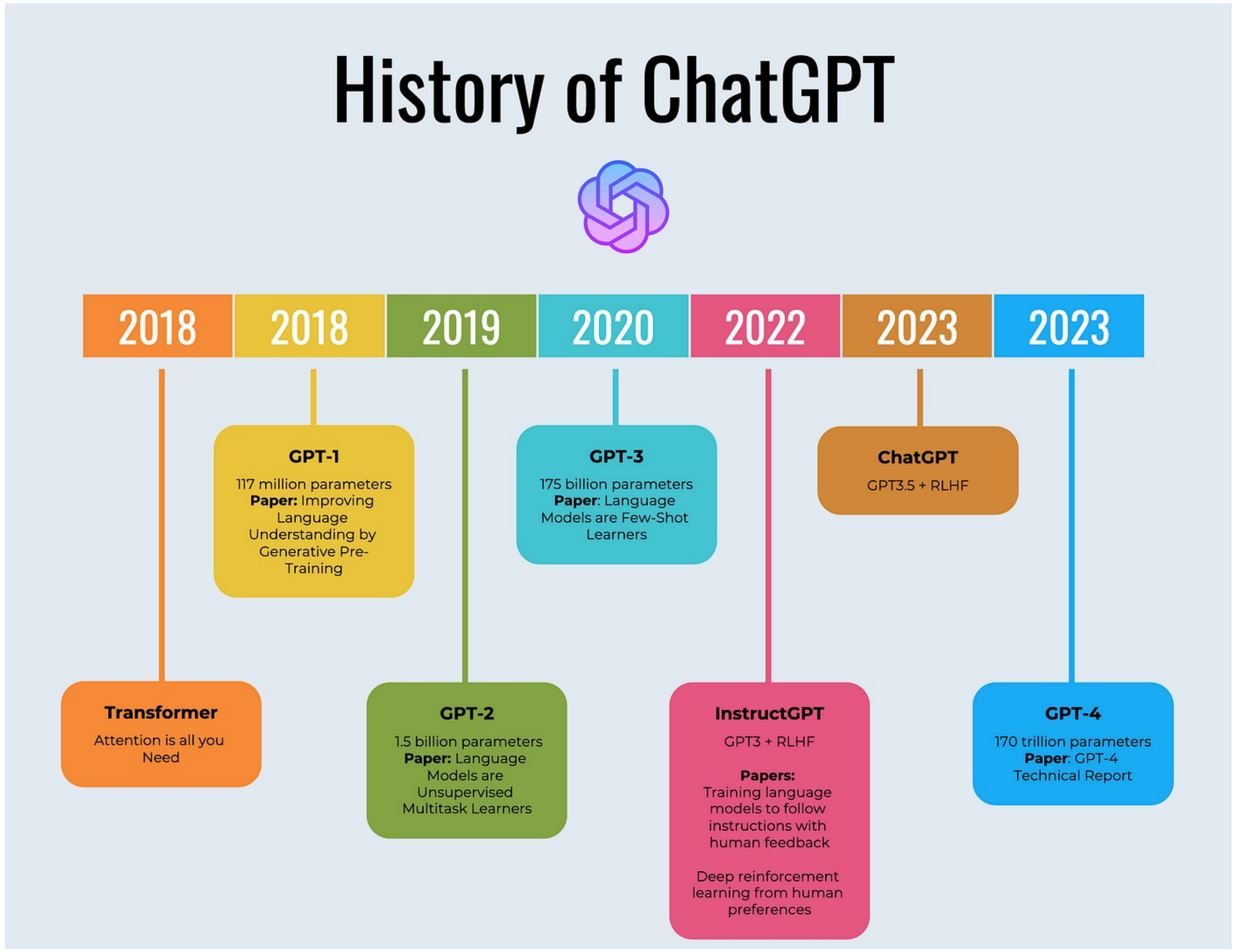
These context words will represent ***banking***

GPT Models

GPT

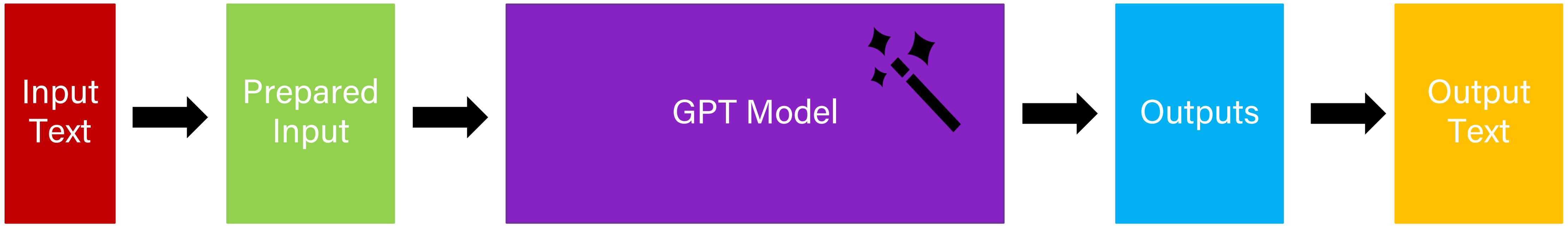
- **G**enerative **P**re-trained **T**ransformer
  - Uni-directional
  - Draws from corpus of information to generate best results for query
  - Based on transformer architecture (decoder only)

Timeline of GPT Models

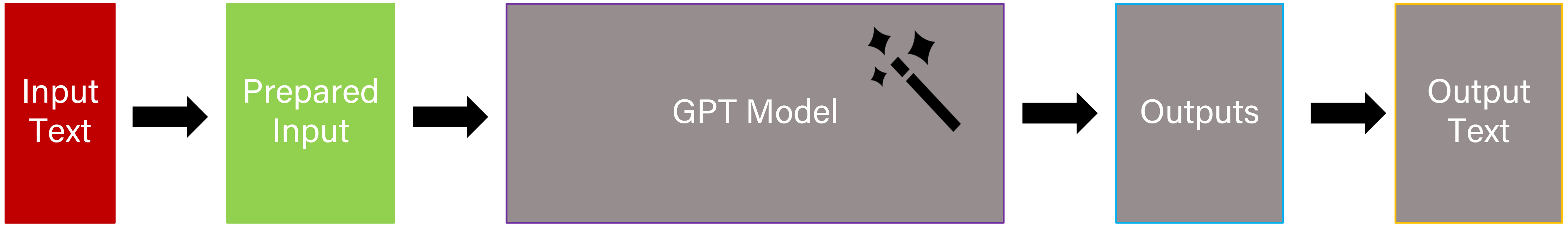

History of ChatGPT

2018	2018	2019	2020	2022	2023	2023
Transformer Attention is all you Need	**GPT-1** 117 million parameters **Paper:** Improving Language Understanding by Generative Pre-Training	**GPT-2** 1.5 billion parameters **Paper:** Language Models are Unsupervised Multitask Learners	**GPT-3** 175 billion parameters **Paper**: Language Models are Few-Shot Learners	**InstructGPT** GPT3 + RLHF **Papers:** Training language models to follow instructions with human feedback; Deep reinforcement learning from human preferences	**ChatGPT** GPT3.5 + RLHF	**GPT-4** 170 trillion parameters **Paper**: GPT-4 Technical Report

Architecture – High Level



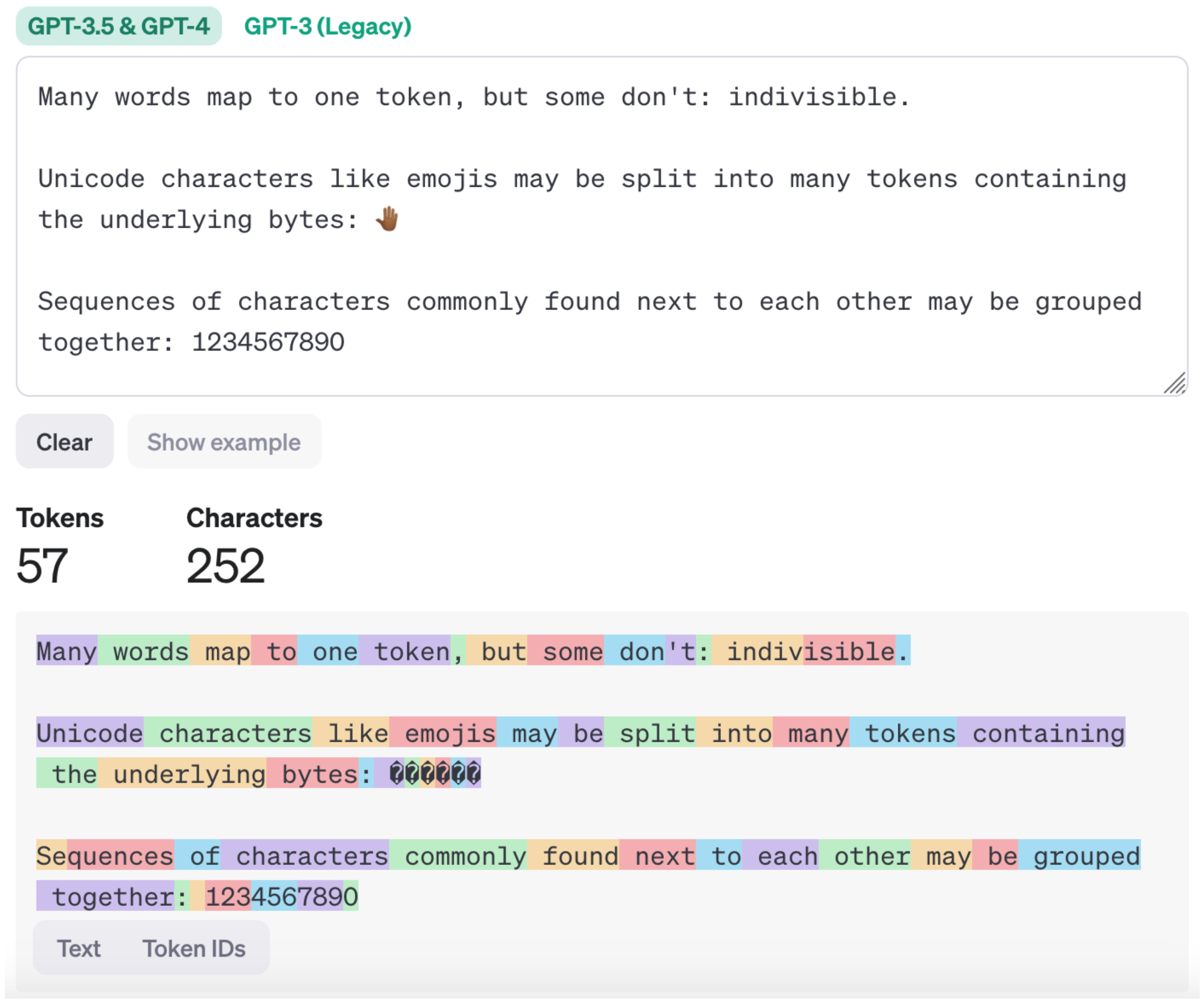
Architecture – High Level



GPT Embeddings

How does this work in GPT?

- Text is broken down into tokens; on average, 75 words ~= 100 tokens
- Learned embedding weights have a dimensionality of 12,288



Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🤚🏾

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Tokens 57 Characters 252

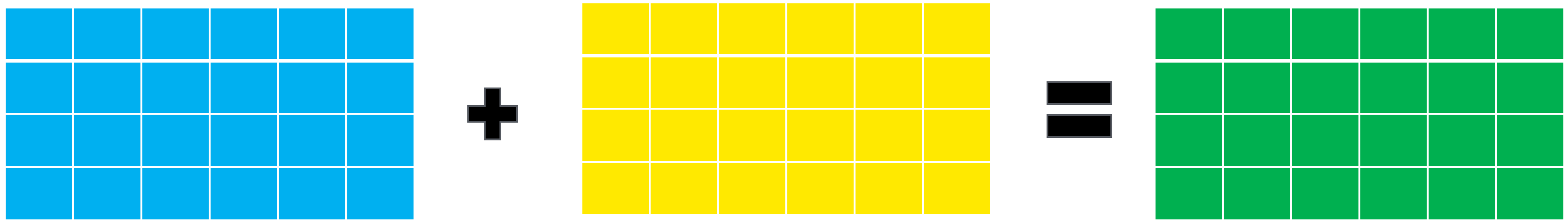
Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: ������

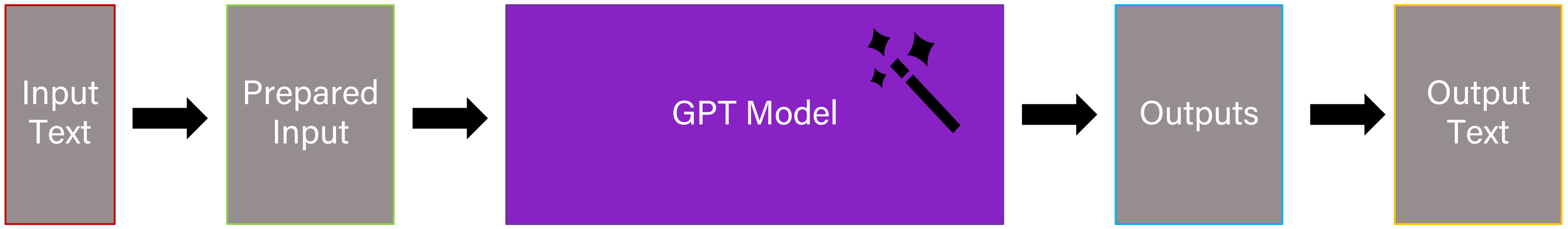
Sequences of characters commonly found next to each other may be grouped together: 1234567890

How does this work in GPT?

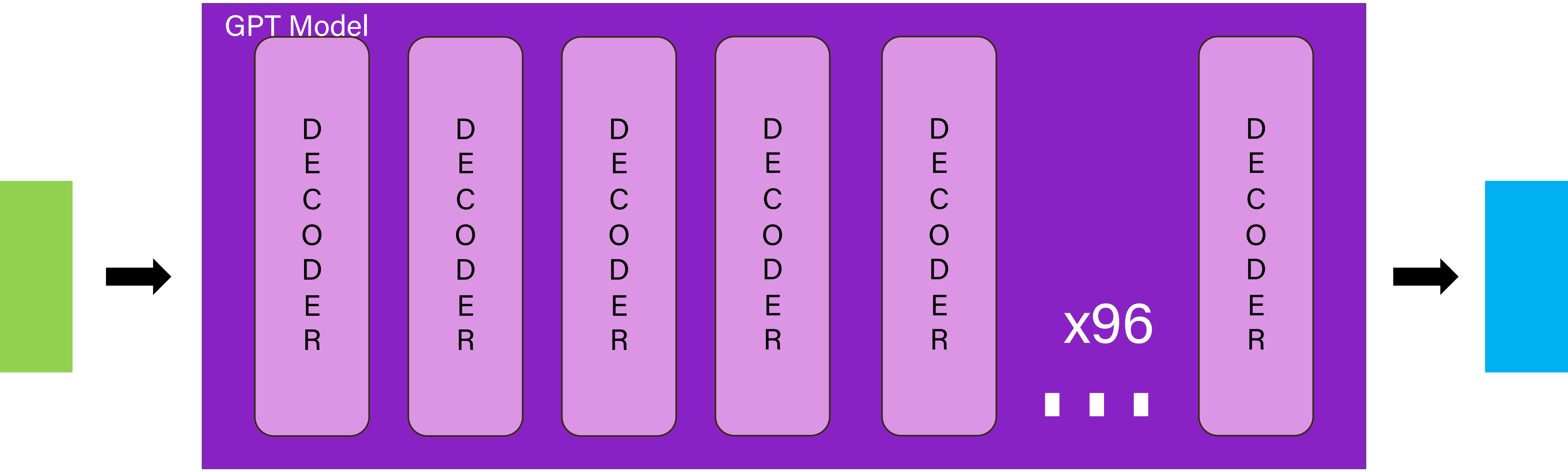
- A **sequence embedding** is represented as a matrix of input sequence length (2048) x 12,288
- **Position embeddings** are also created for each token in the input sequence, yielding another 2048 x 12,288 matrix
- The sequence embedding matrix and position embedding matrix are summed to yield the final input matrix



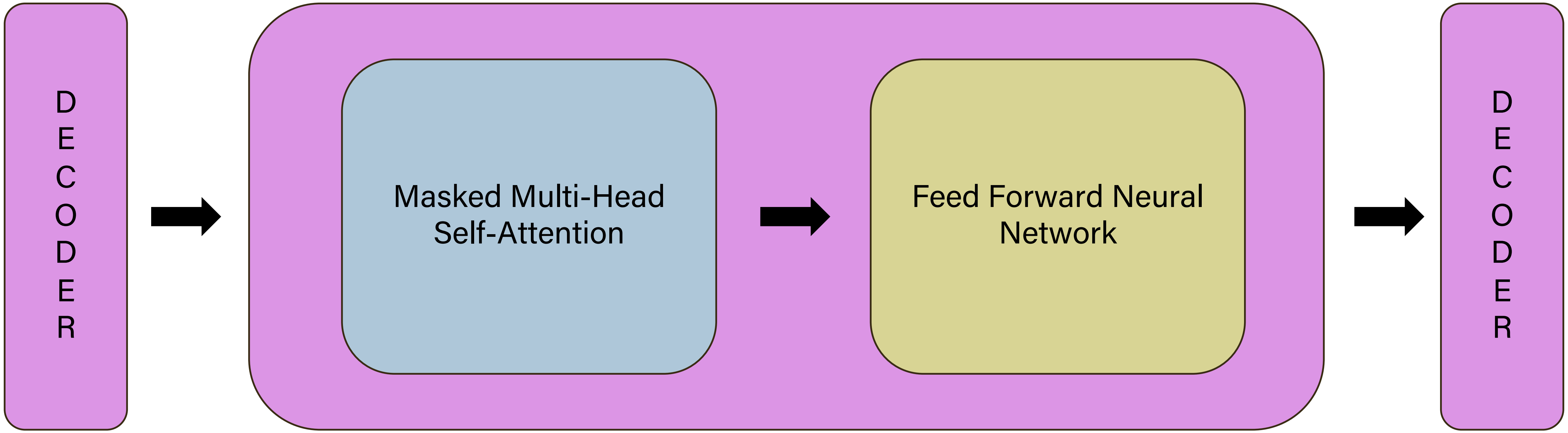
Architecture – High Level



Architecture – High Level



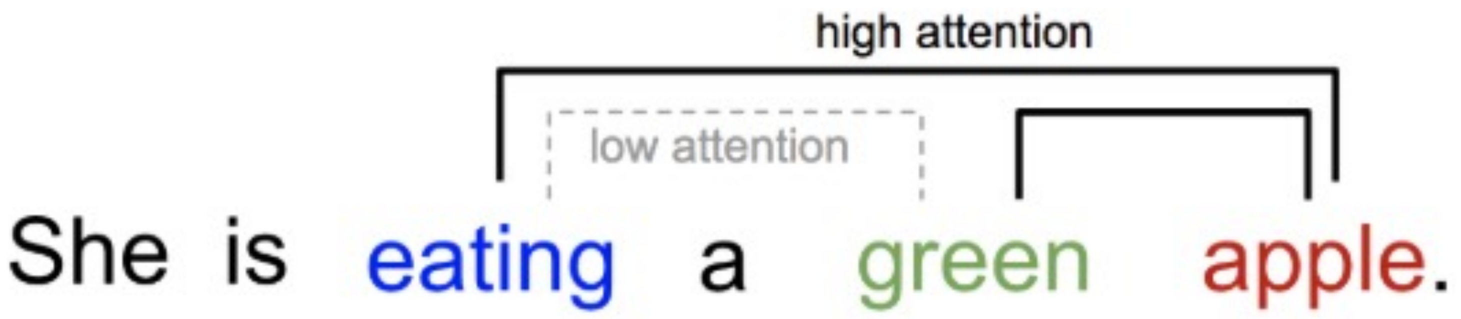
Architecture – High Level



Attention

What is attention?

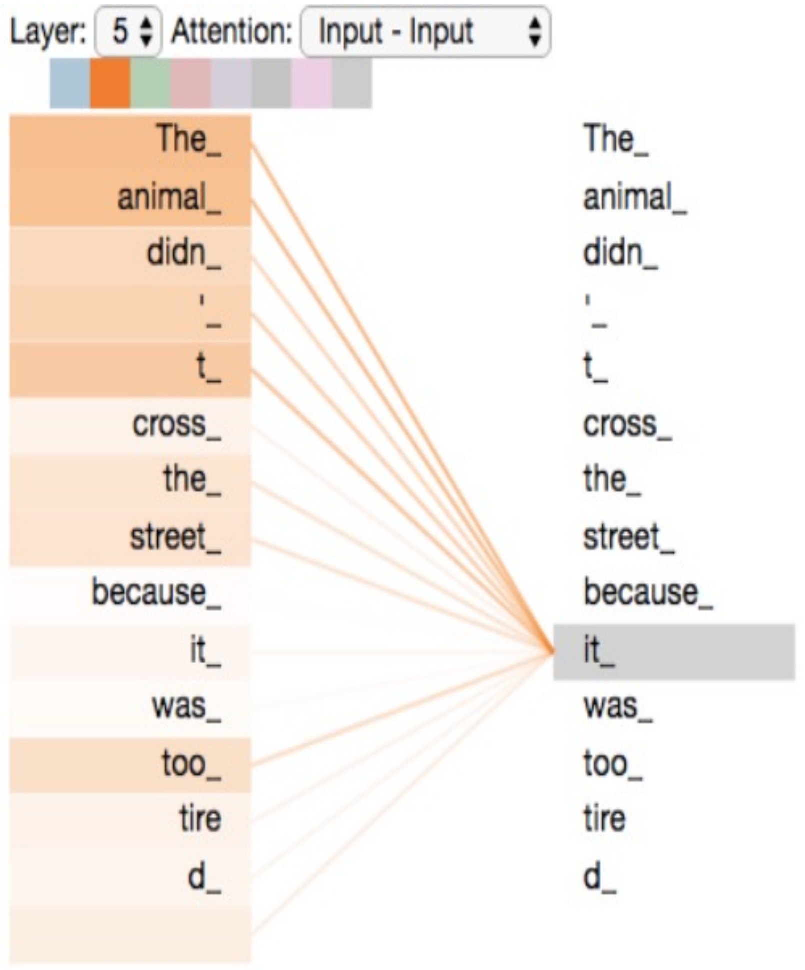
To predict a word, use only the most relevant parts of the input text



Attention

Self-attention: relating different positions of a single sequence to itself to compute attention

1. Score how relevant every other token in the sequence is to the token being processed
2. Softmax the scores to normalize them
3. Calculate a "weighted average" vector representation by multiplying each context token's embedding by its score and summing



Purdue University Rosen Center for Advanced Computing

Attention

Self-attention under the hood:

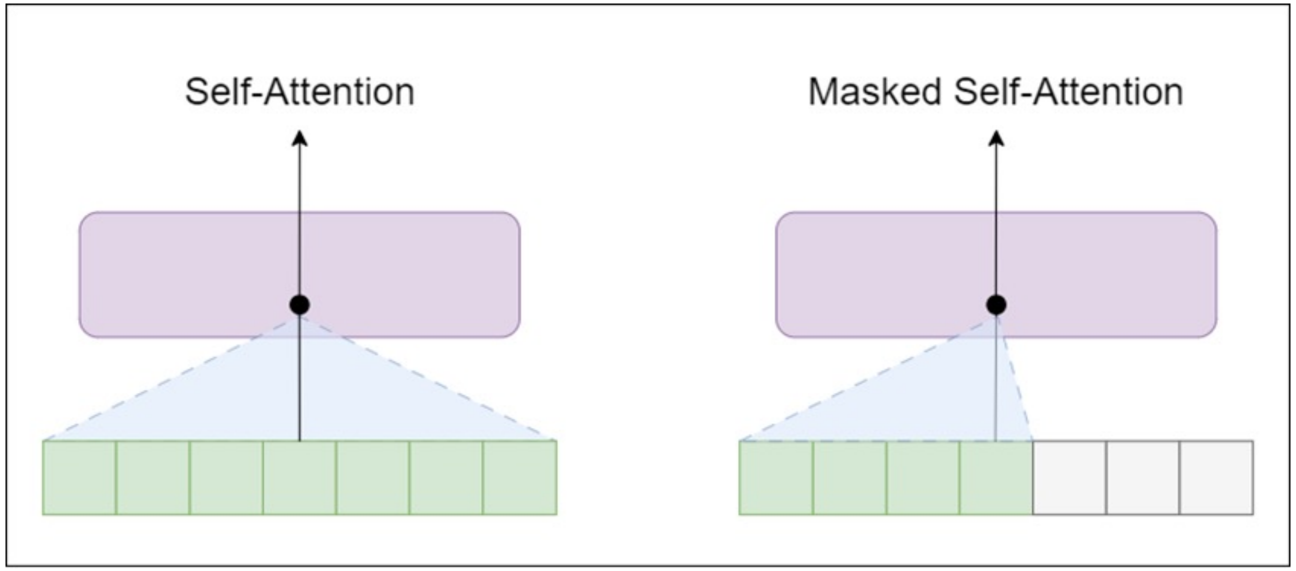
- Based on query, key, and value vectors
  - Derived by multiplying the input token embedding by W^Q, W^K, and W^V matrices that are learned during training
- Query is a representation of the current token
- This is multiplied with the key vectors of all tokens in the input sequence to create a score for each token
- This score is normalized, then multiplied with the value vectors of each other token
- The results are summed

$$\boxed{SelfAttention(X) = \sum_j Attention(q_i k_j) v_j}$$

Attention

Masked Self-Attention: prevents the model from making decisions on future information

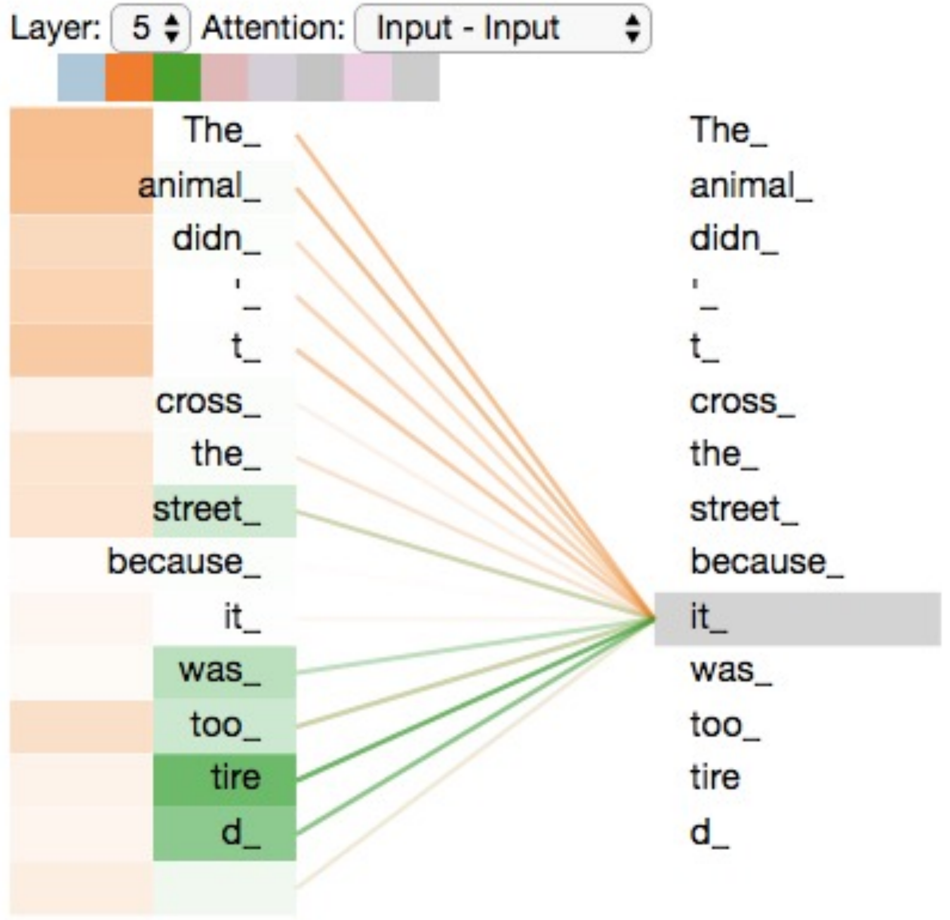
- Do not allow information from future tokens to impact the representation of the current token (scores for future tokens are -∞)
- Maintains the "auto-regressive" nature of GPT



Attention

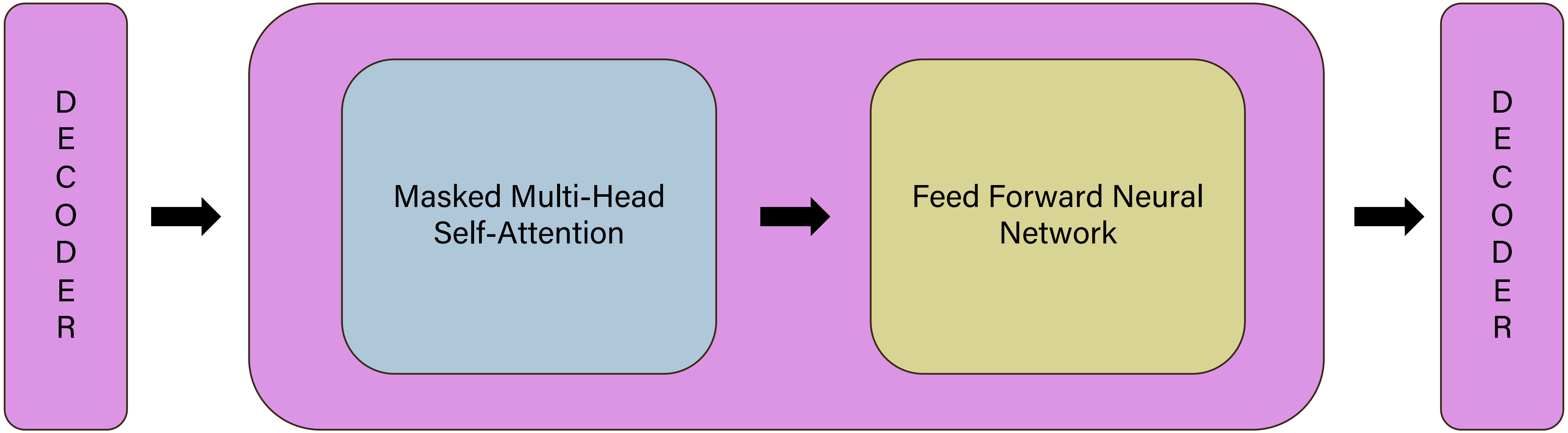
Multi-Headed Self-Attention:
Self-attention, but performed *n* times in parallel to build a better representation

- Use different query, key, and value vectors for each head
- Creates richer resultant representation of the input



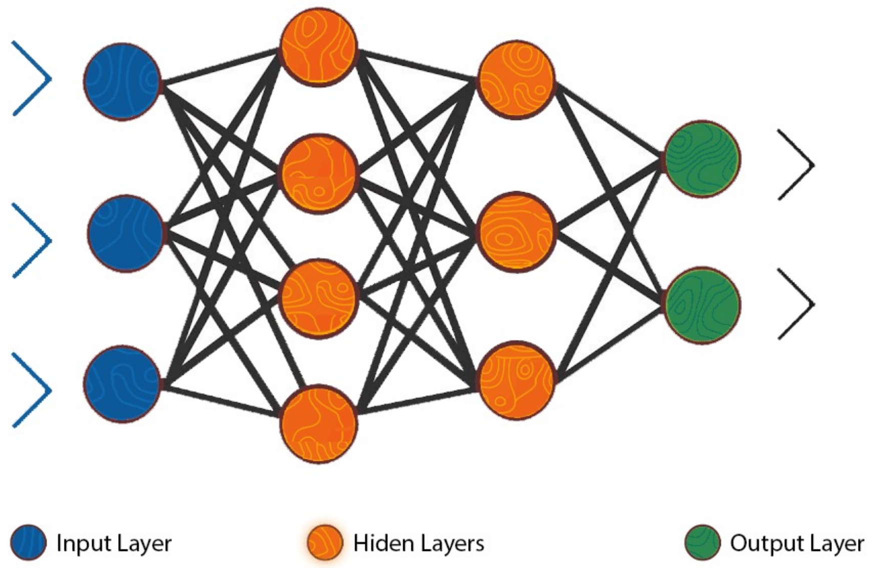
Rosen Center for Advanced Computing

Architecture - High Level

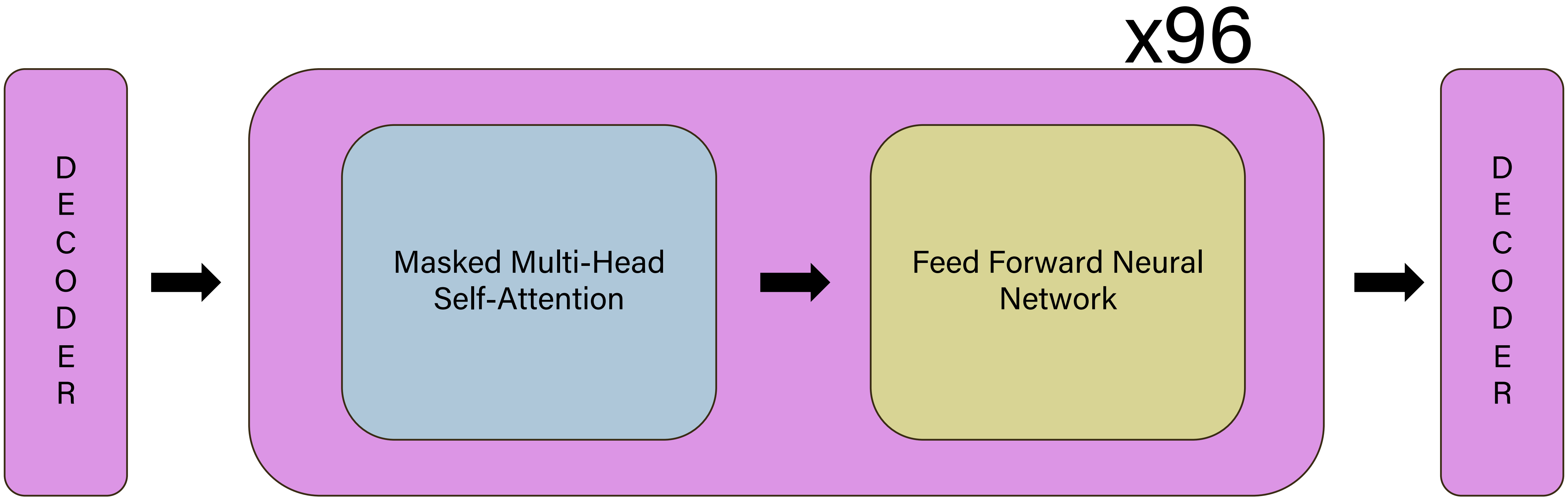


Feed Forward Network

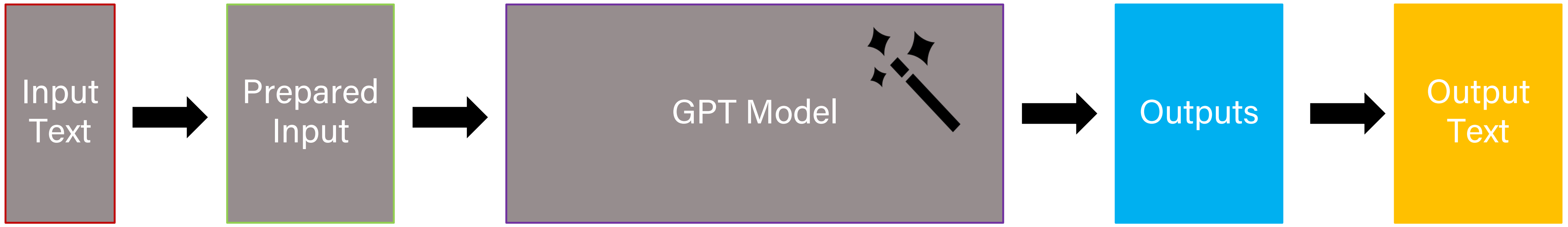
- Processes the input representation (which includes context from the attention mechanism) to generate output
  - Output is either passed to the next decoder or is used to generate output in the case of the last decoder
- Simple multi-layer perceptron



Architecture – High Level

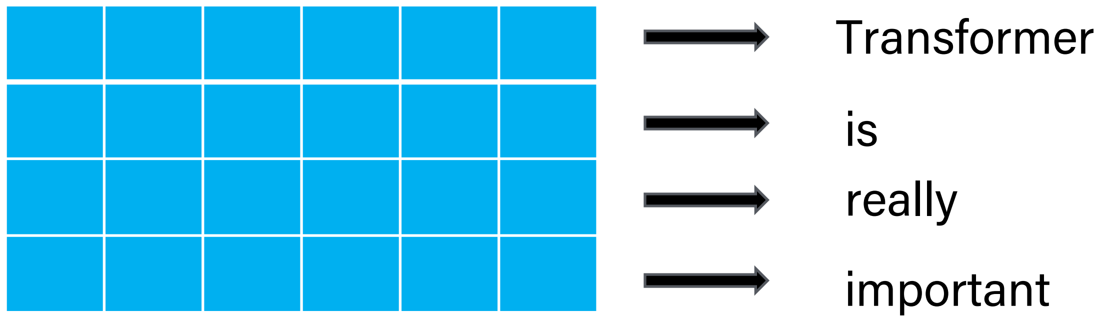


Architecture – High Level



Converting output to words

- Reverse the process used at the beginning:
  - Model outputs a 2048 x 12,288 matrix
  - Multiply this with the inverse of the one-hot encoding word embedding matrix to get output token probabilities
  - Apply softmax
  - Get output text!

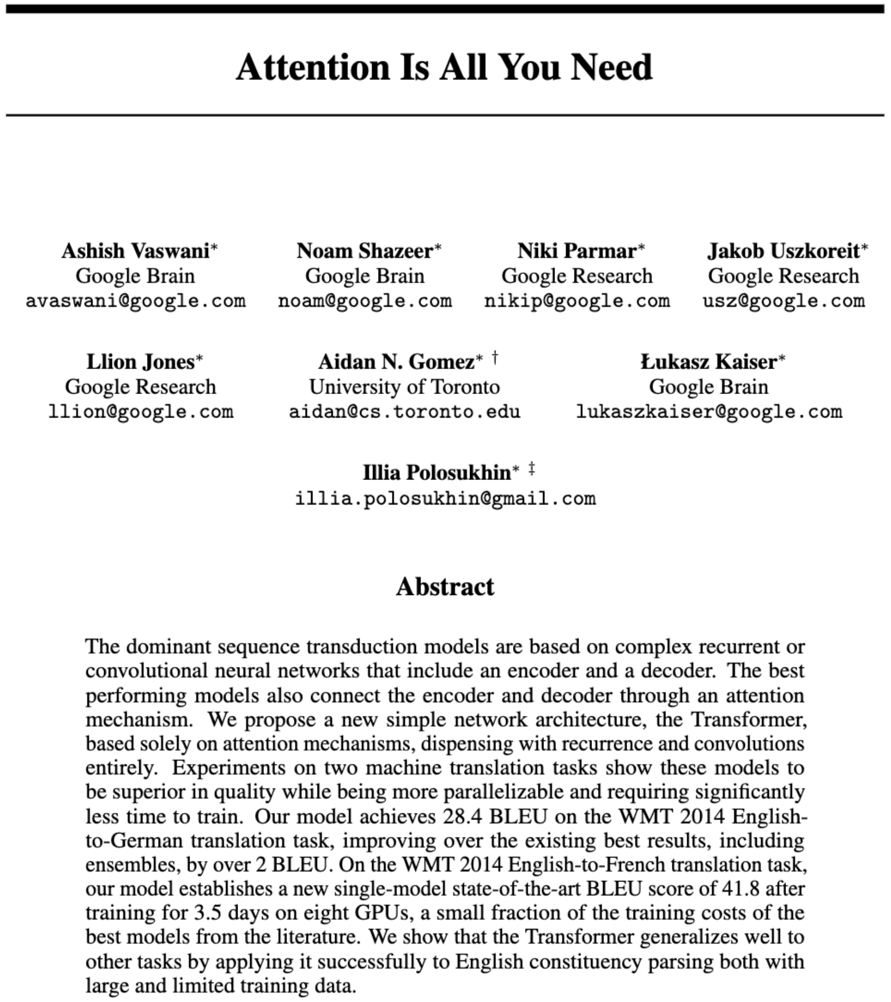


Review of Key NLP Terms

- **Embedding**: way to numerically represent the meaning of a word, sentence, paragraph, etc.
- **Language Model**: probabilistic model of words and phrases in a language
- **Transformers**: Architecture based on attention mechanisms
- **Self-Attention:** A method of “baking in” context to the representation of a token based on the relative importance of each context token

Takeaways

- The ”magic” of modern NLP comes down to two things: the volume of data and the number of parameters that can be trained
- Despite the hype around GPT, the field has not fundamentally changed since Transformers
- Understanding “Attention is all you need” gets you 90% of the way to understanding GPT



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

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.