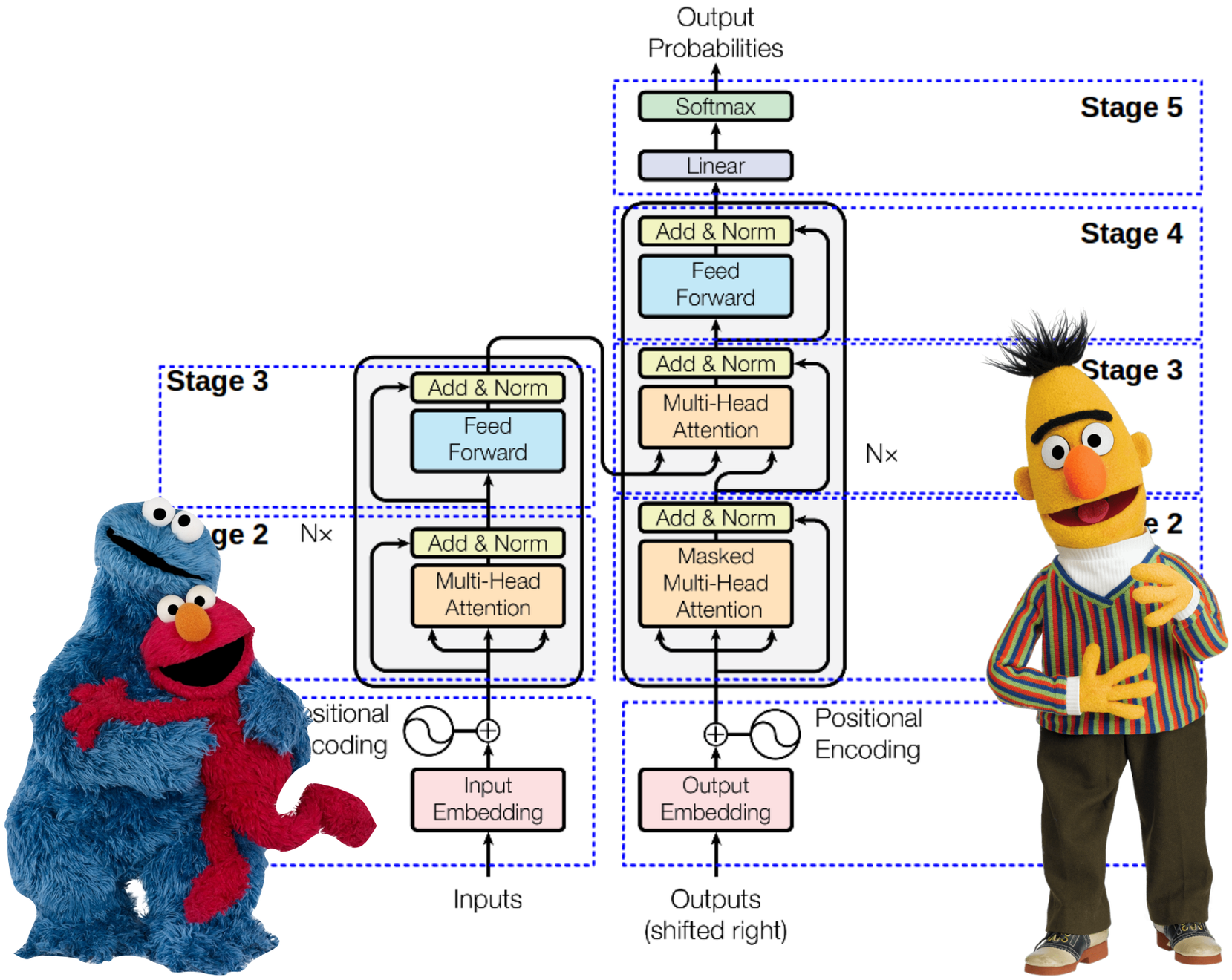
Context-based models

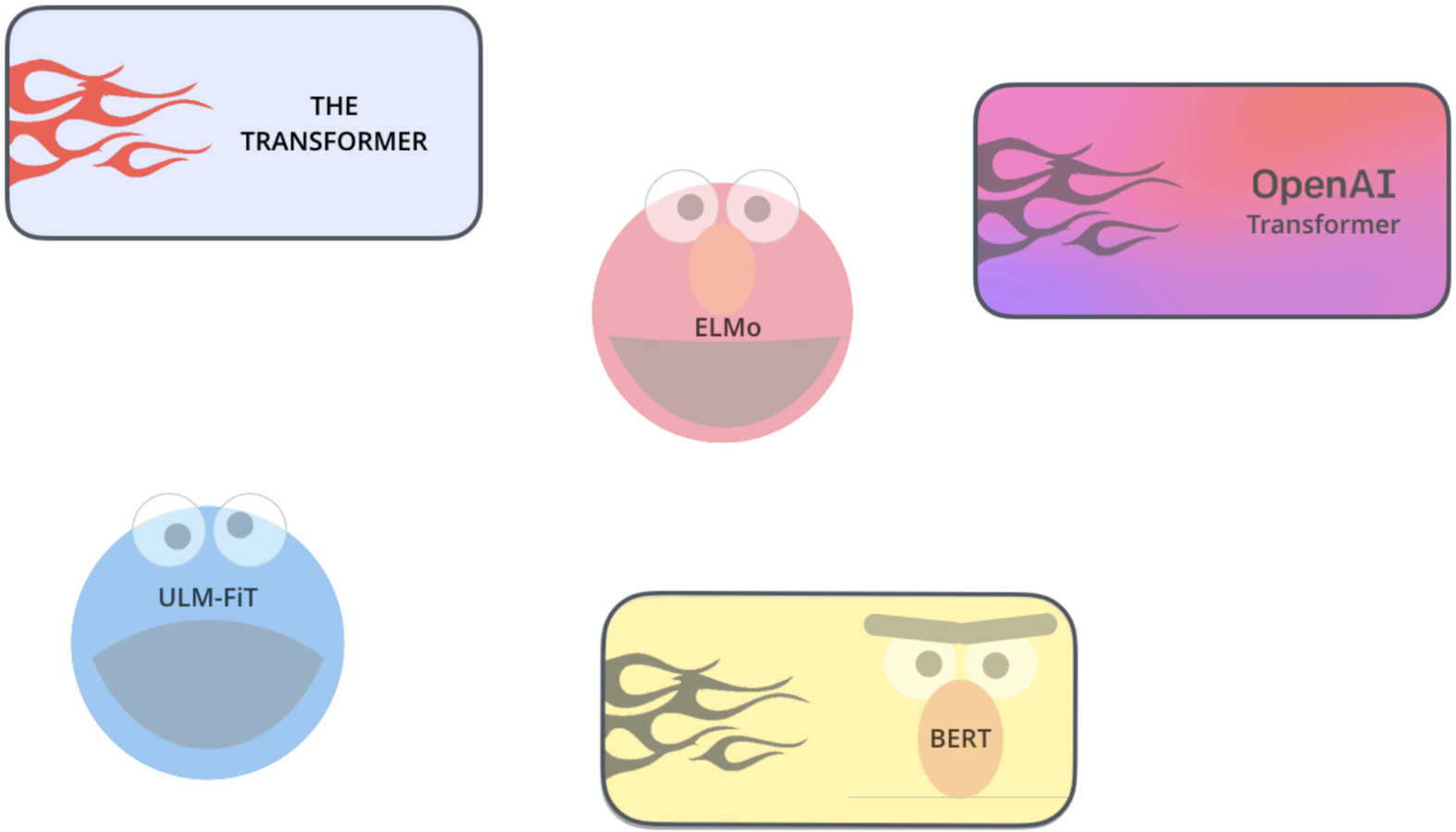
ELMo, BERT, GPT

Кафедра технологий проектирования сложных технических систем

2025


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Positional Encoding
Output Embedding
Outputs (shifted right)
Stage 5
Stage 4
Stage 3
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Input Embedding
Inputs
Stage 3
Stage 2

Другие модели



The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

План лекции

The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

ELMo

1. Expedited Labour Market Opinion
2. Electric Light Machine Organization
3. Enough Let’s Move On

Special thx for @Anastasia Yanina:

https://github.com/ml-mipt/ml-mipt/blob/advanced/week05_BERT_and_LDA/Lecture_BERT_DIHT.pdf

ELMo

1. Expedited Labour Market Opinion
2. Electric Light Machine Organization
3. Enough Let's Move On

4. Embeddings from Language Models

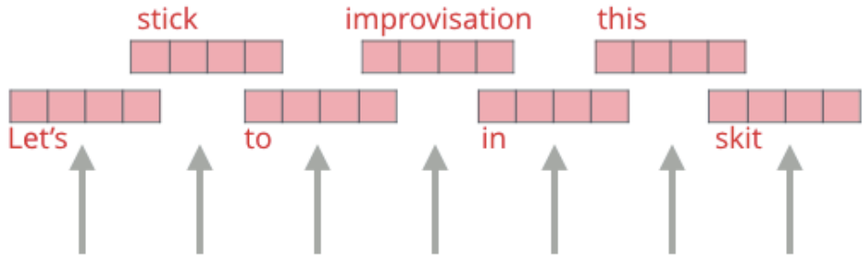
Special thx for @Anastasia Yanina:

https://github.com/ml-mipt/ml-mipt/blob/advanced/week05_BERT_and_LDA/Lecture_BERT_DIHT.pdf


Hey ELMo, what's the embedding of the word "stick"?
There are multiple possible embeddings! Use it in a sentence.
Oh, okay. Here:
"Let's stick to improvisation in this skit"
Oh in that case, the embedding is:
-0.02, -0.16, 0.12, -0.1 ….etc

ELMo

ELMo Embeddings



Words to embed

ELMo

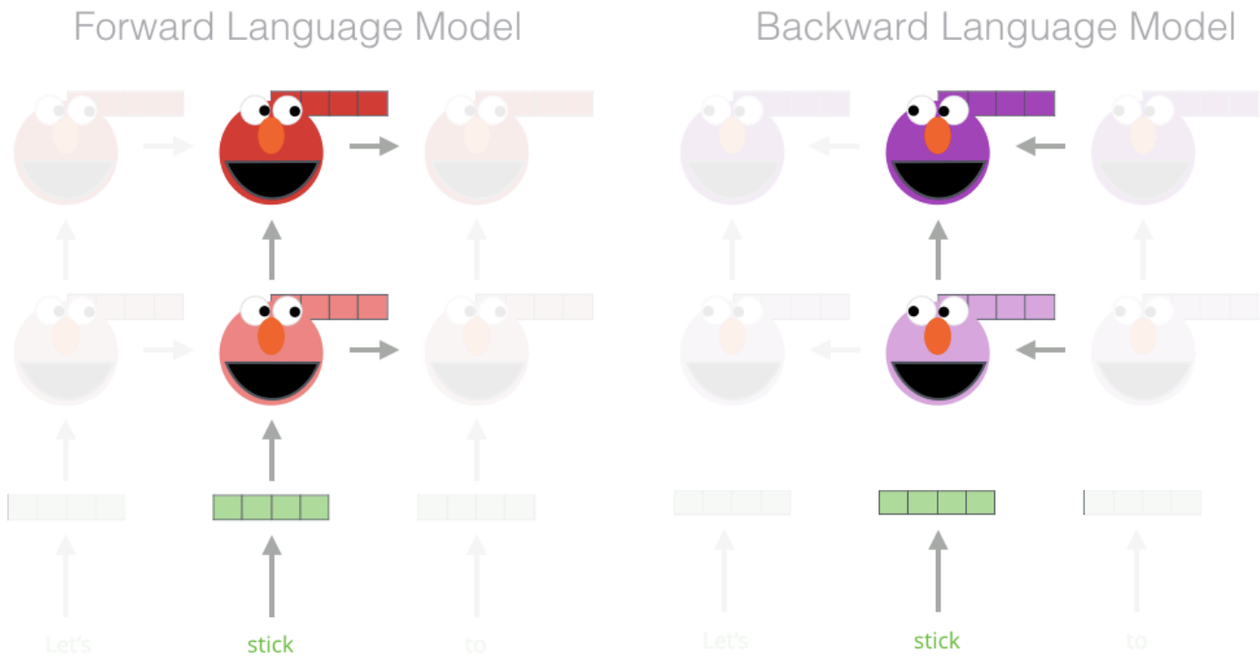
Embedding of “stick” in “Let’s stick to” - Step #2

1- Concatenate hidden layers

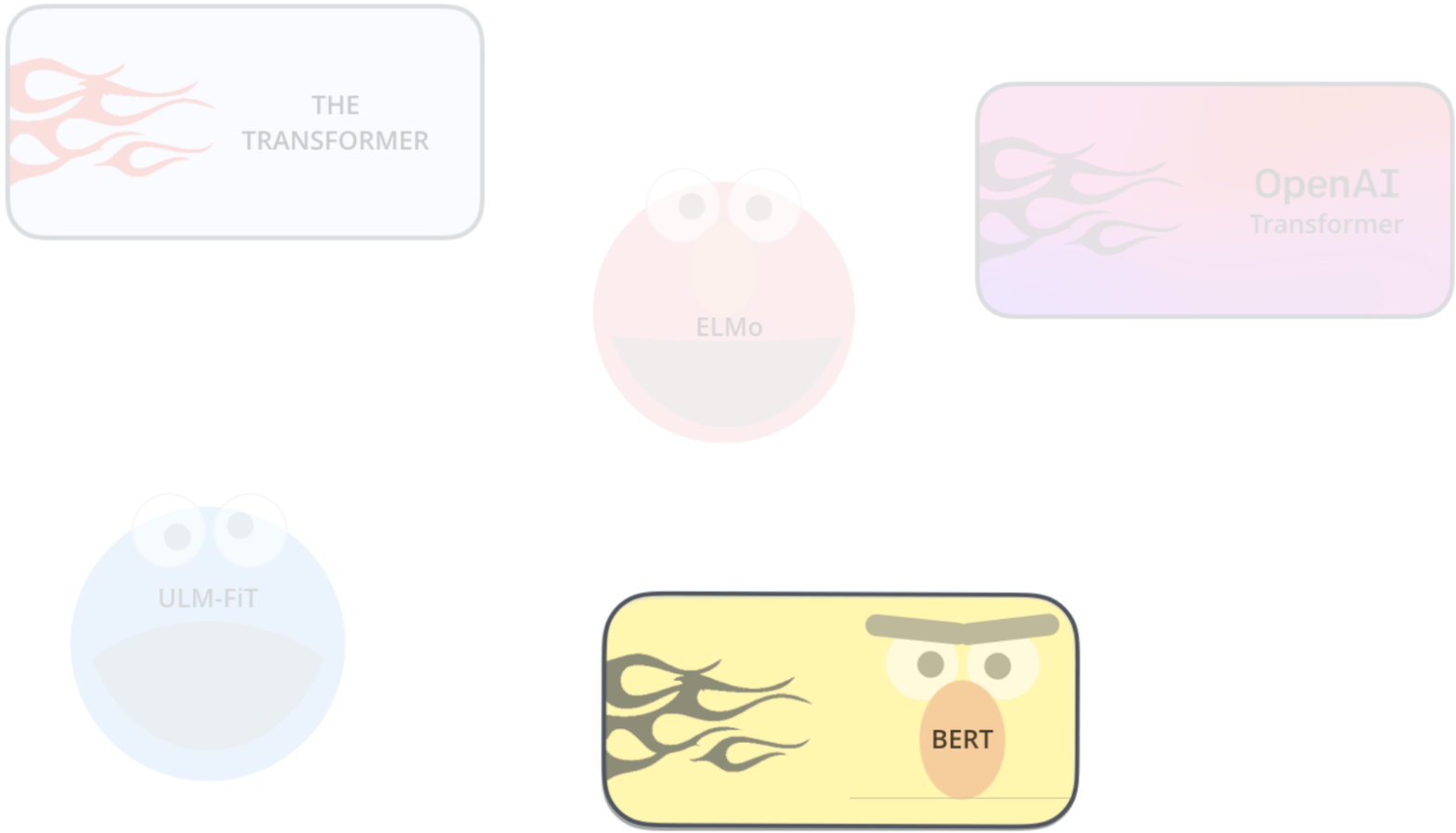
2- Multiply each vector by a weight based on the task

3- Sum the (now weighted) vectors

ELMo embedding of “stick” for this task in this context



Другие модели

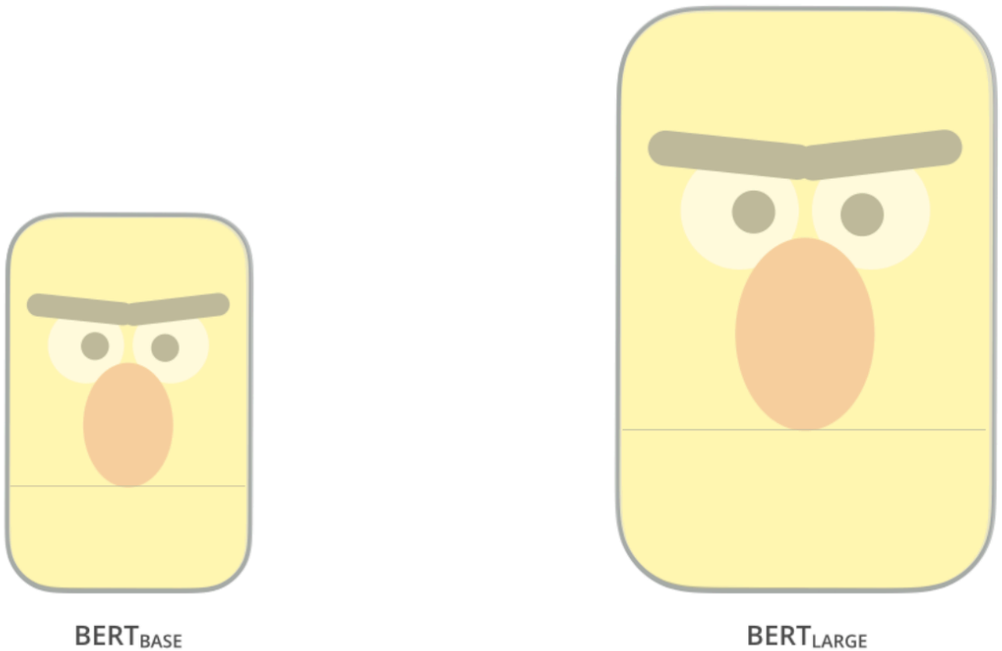


The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

BERT

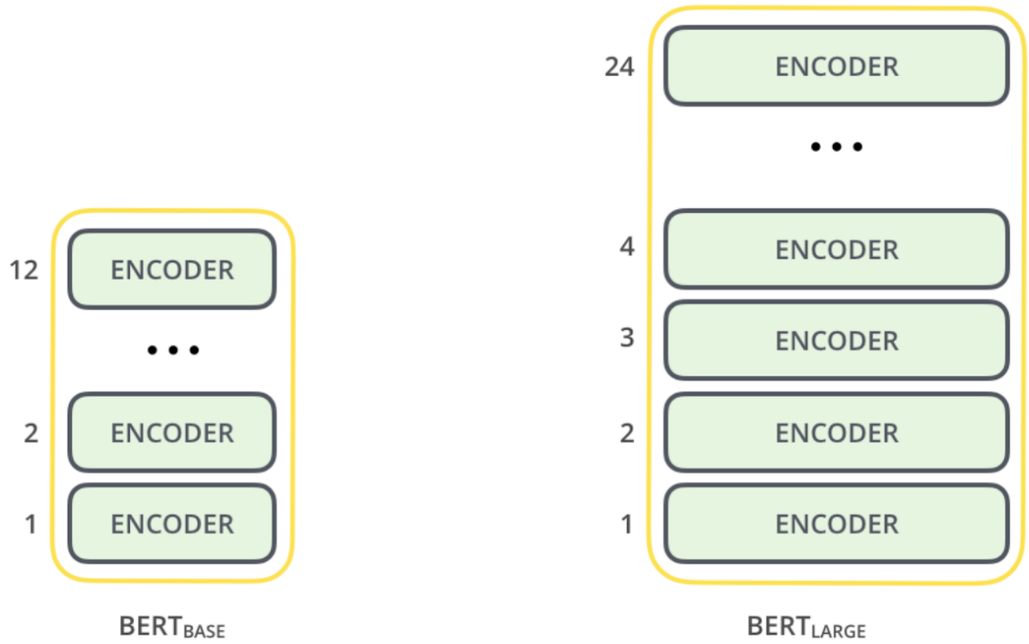
[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT



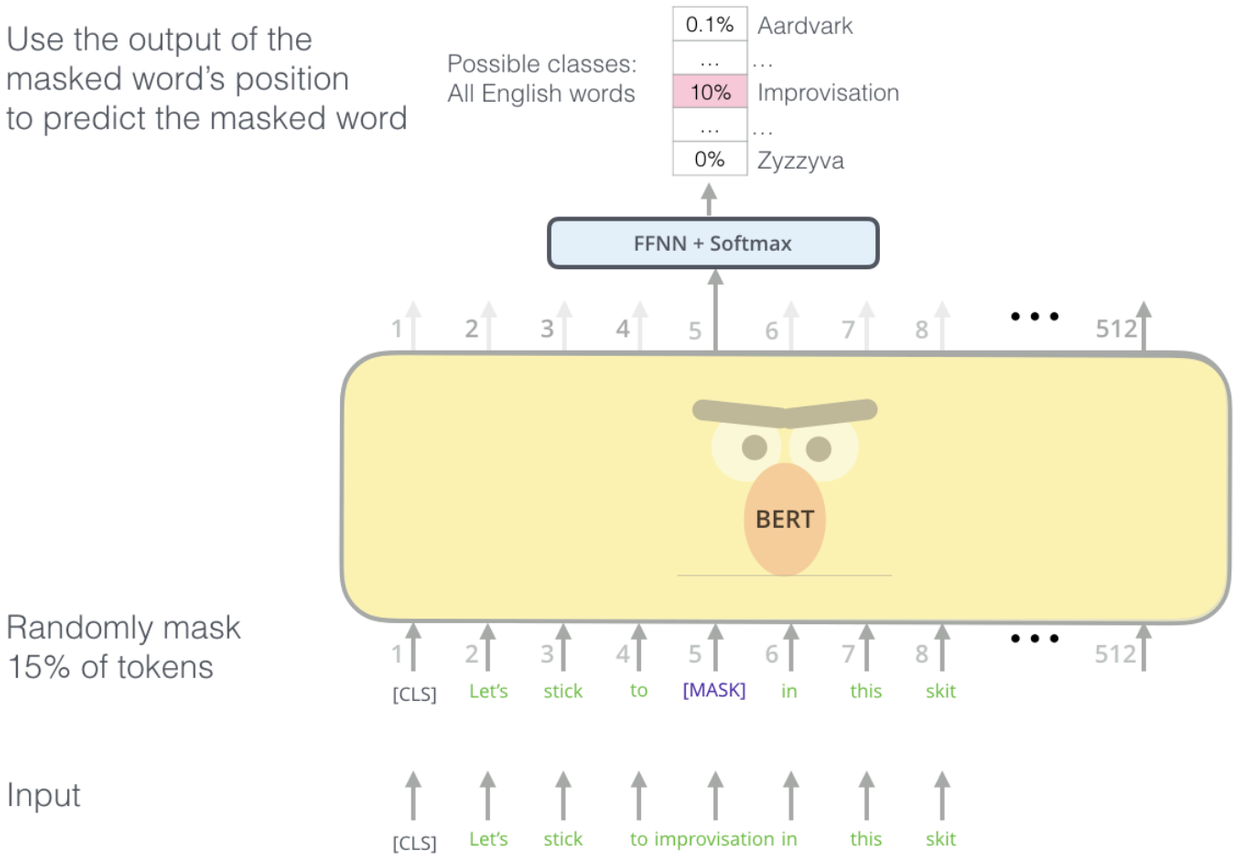
[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT



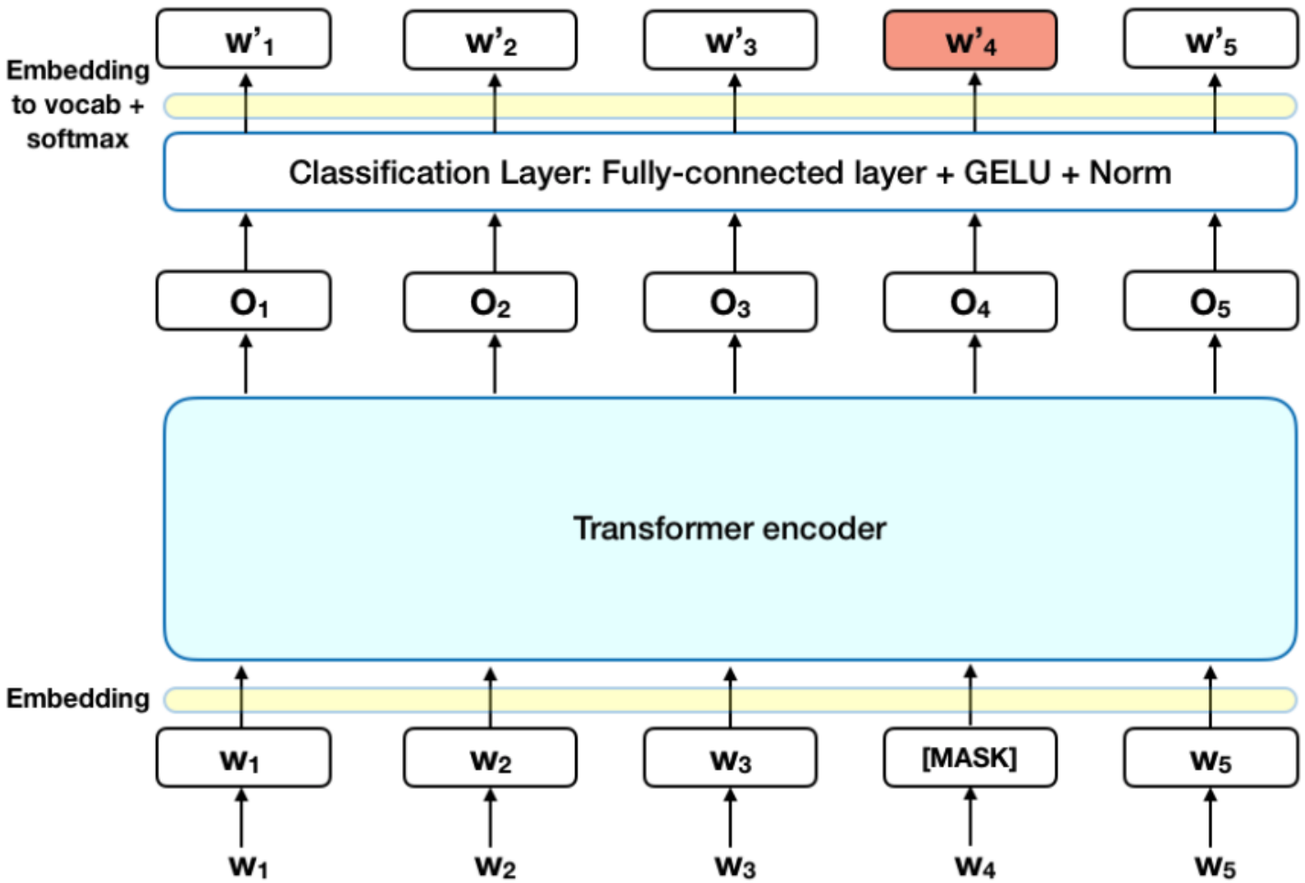
[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT



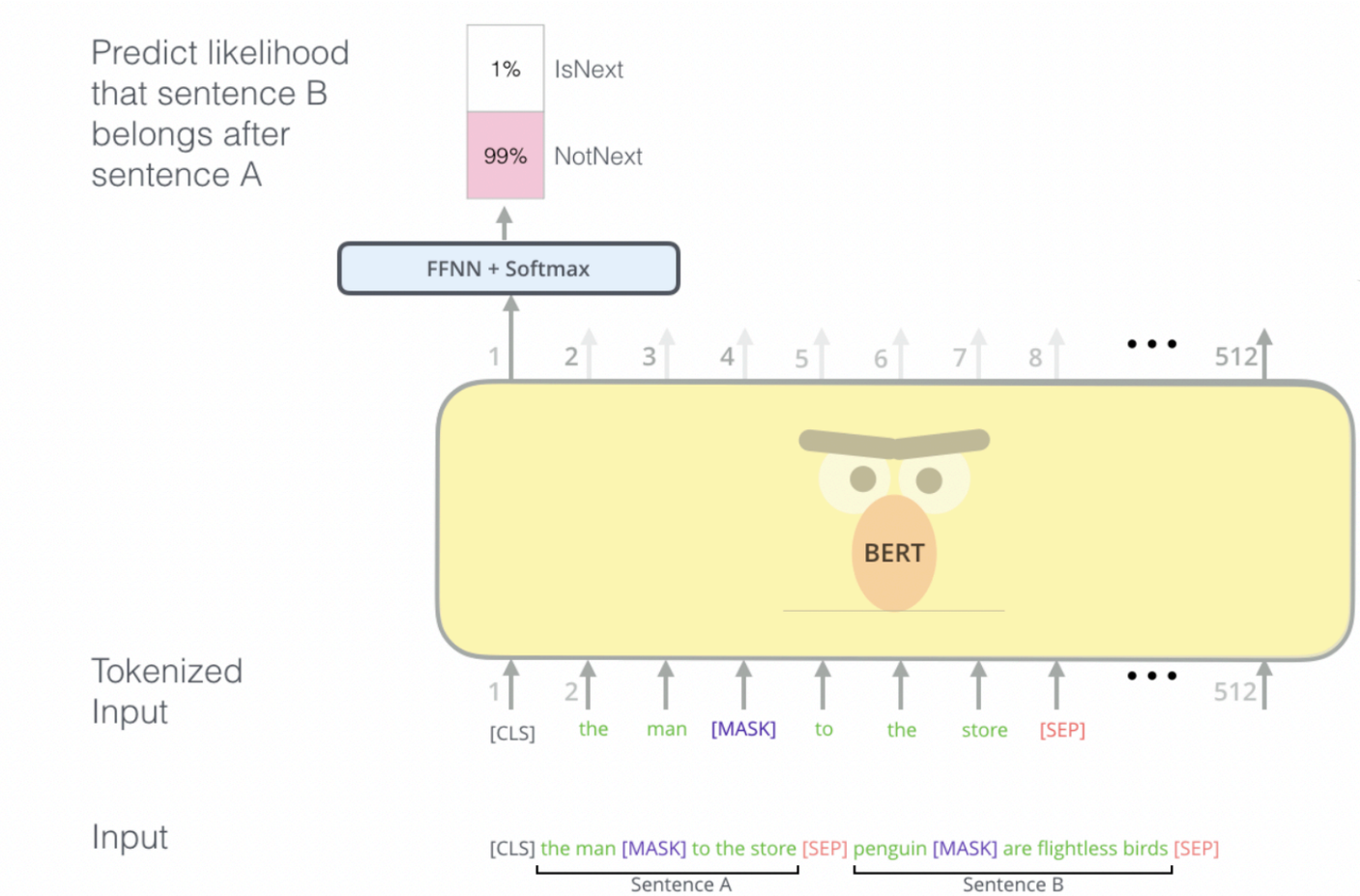
[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT



[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT



The second task BERT is pre-trained on is a two-sentence classification task. The tokenization is oversimplified in this graphic as BERT actually uses WordPieces as tokens rather than words --- so some words are broken down into smaller chunks.

[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Byte Pair Encoding

this is the hugging face course. this chapter is about tokenization. this section shows several tokenizer algorithms.

this | is | the | hugging | face | course | . | this | chapter | is | about | tokenization | . | this | section | shows | several | tokenizer | algorithms | .

3x this | 2x is | 1x the | 1x hugging | 1x face | 1x course | 3x . | 1x chapter | 1x about | 1x tokenization | 1x section | 1x shows | 1x several | 1x tokenizer | 1x algorithms

Corpus

1	huggingface
1	hugging
1	face
1	hug
1	hugger
2	learning
2	learner
2	learners
1	learn

Splits

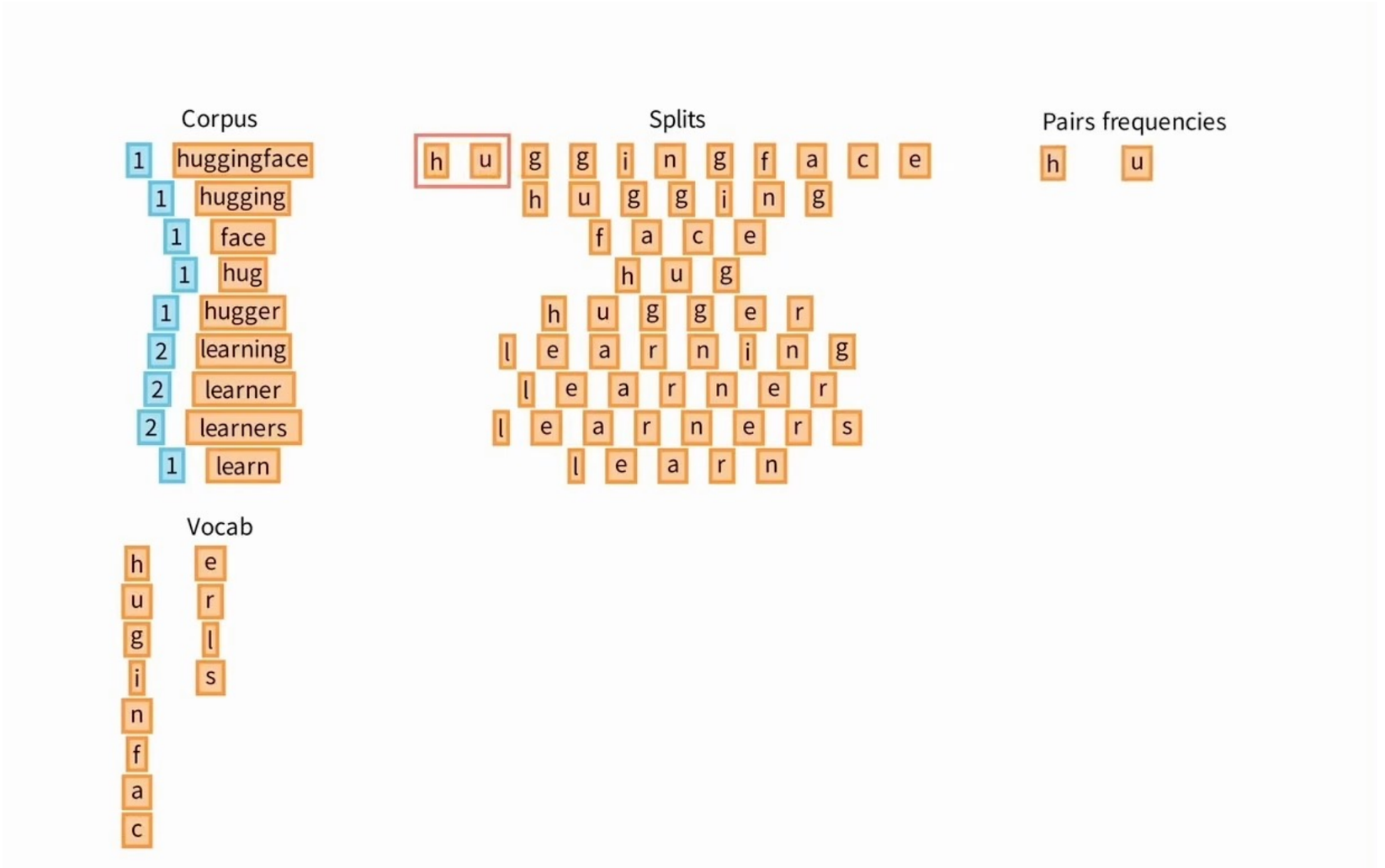
h u g g i n g f a c e
h u g g i n g
f a c e
h u g
h u g g e r
l e a r n i n g
l e a r n e r
l e a r n e r s
l e a r n

Vocab

h u g i n f a c e r l s

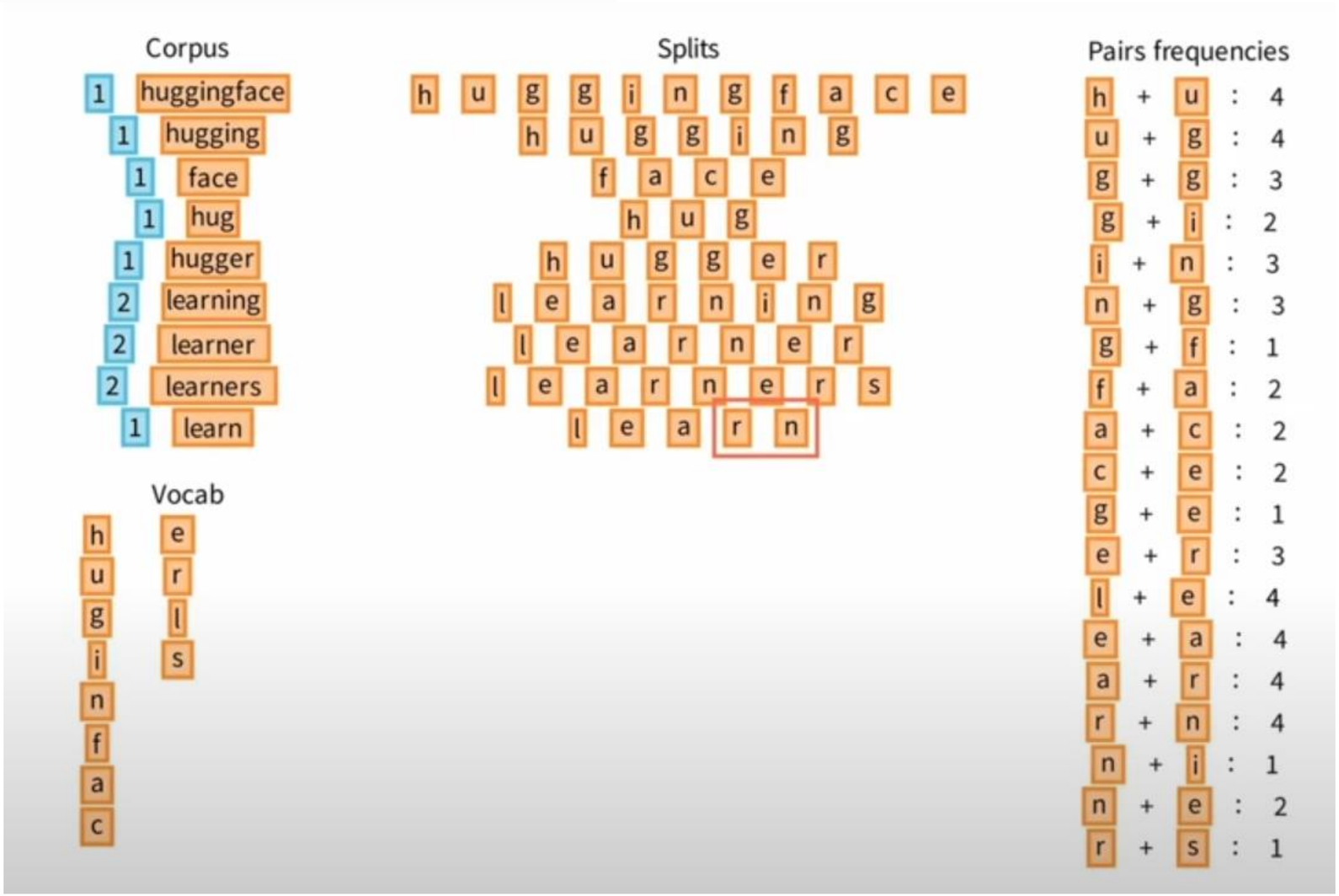
Byte Pair Encoding (BPE) *Byte-Pair Encoding tokenization (HuggingFace)*

Byte Pair Encoding

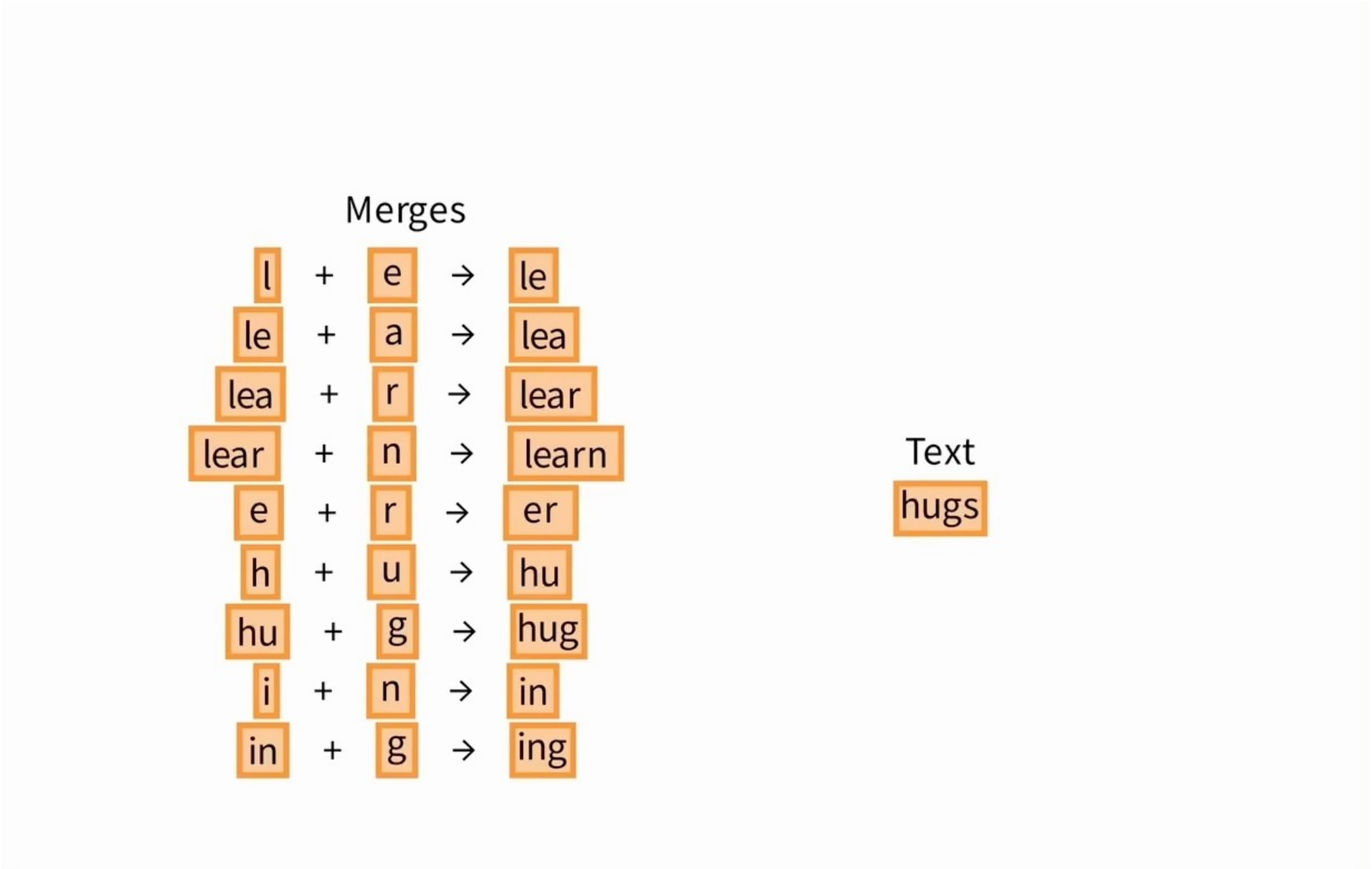


Byte-Pair Encoding tokenization (HuggingFace)

Byte Pair Encoding

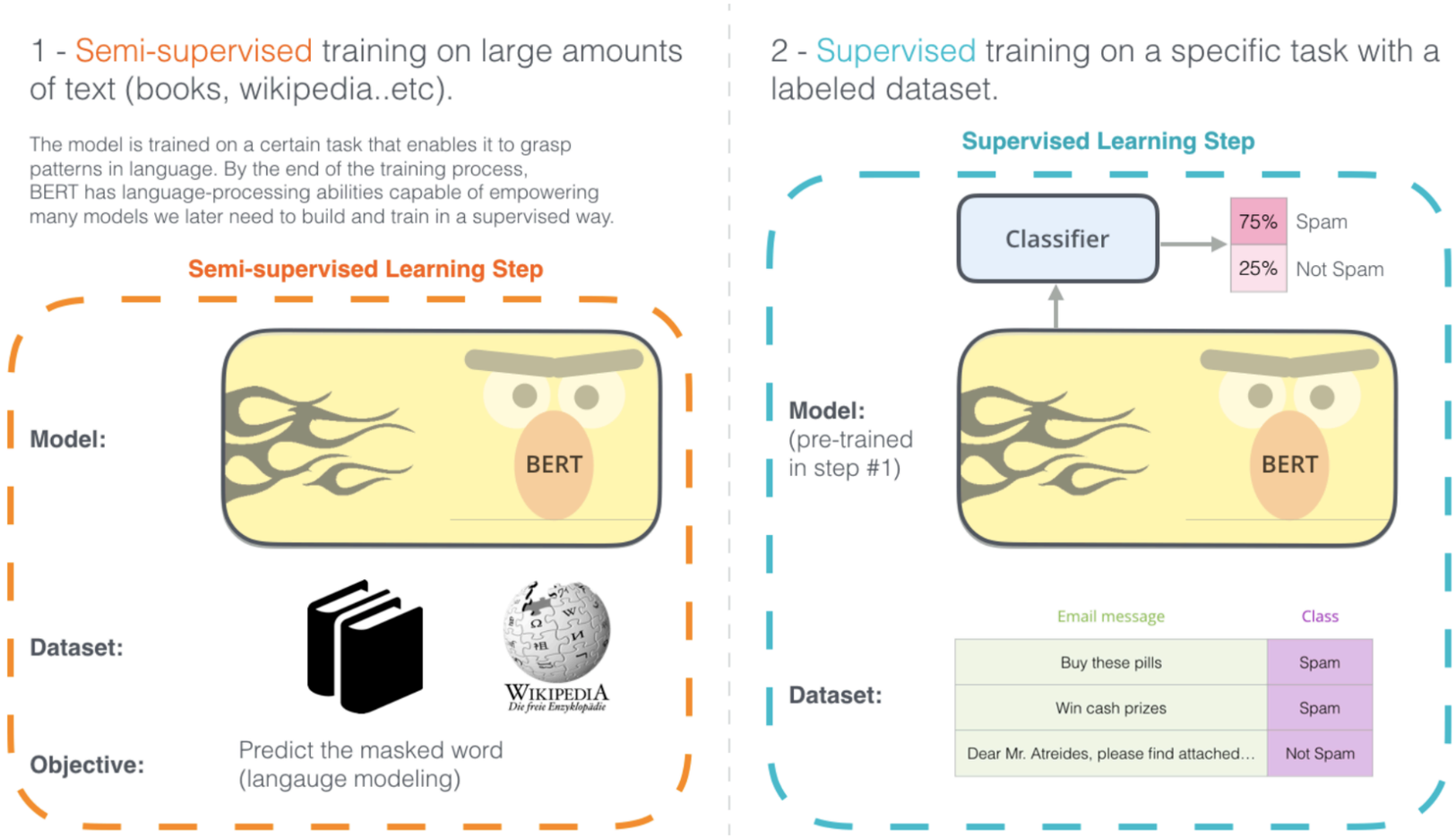


Byte Pair Encoding



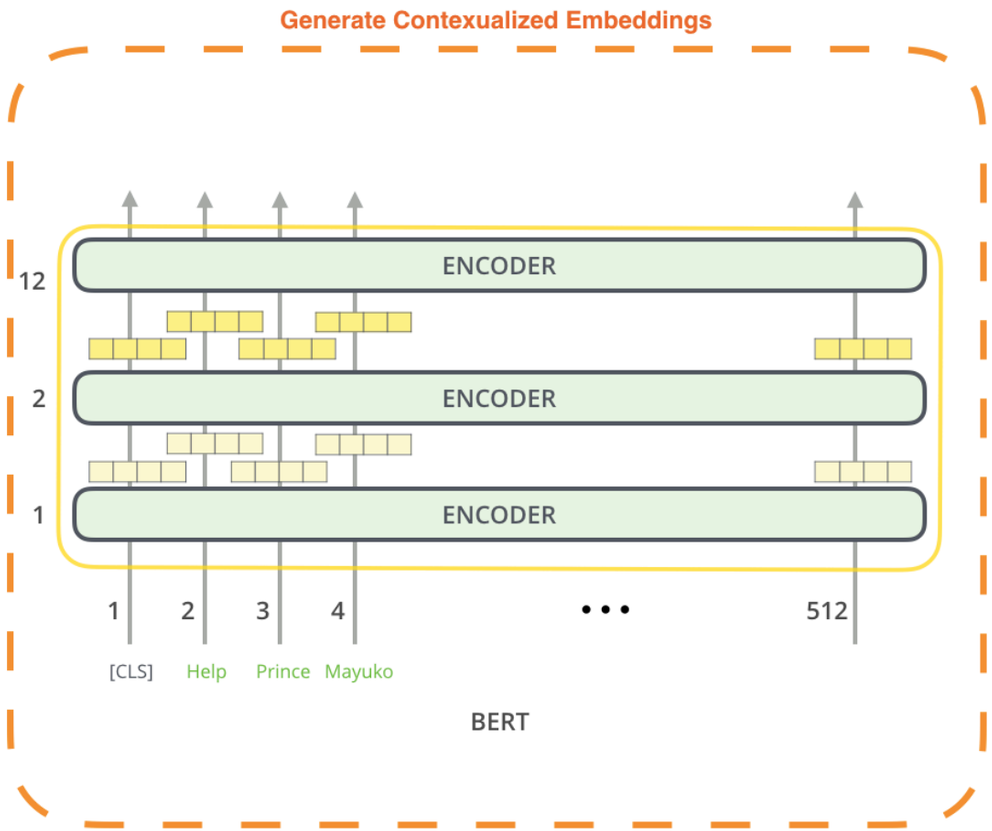
Byte-Pair Encoding tokenization (HuggingFace)

BERT

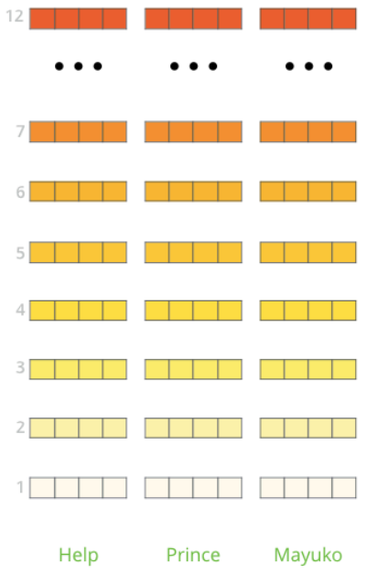


[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT for feature extraction



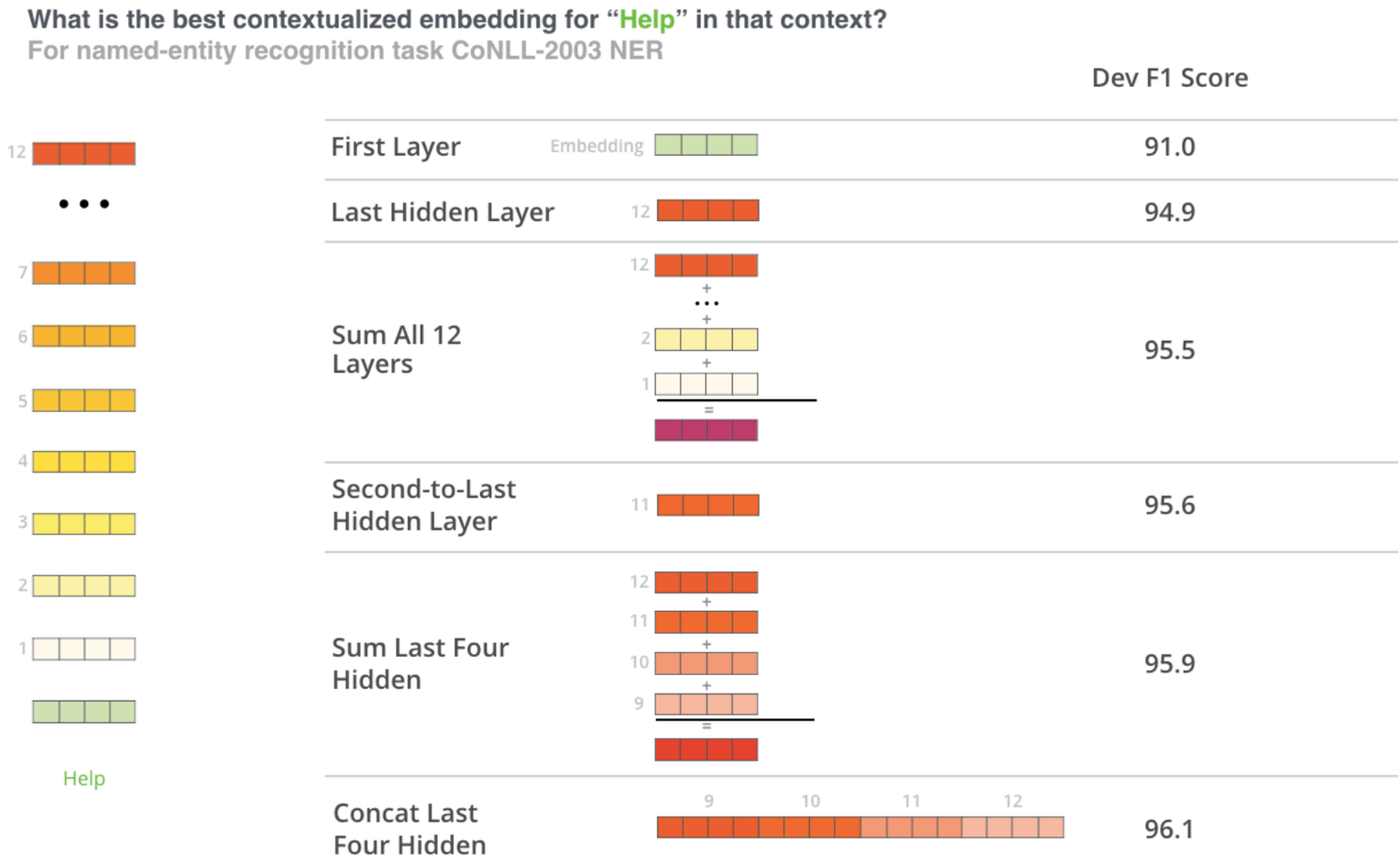
The output of each encoder layer along each token's path can be used as a feature representing that token.



But which one should we use?

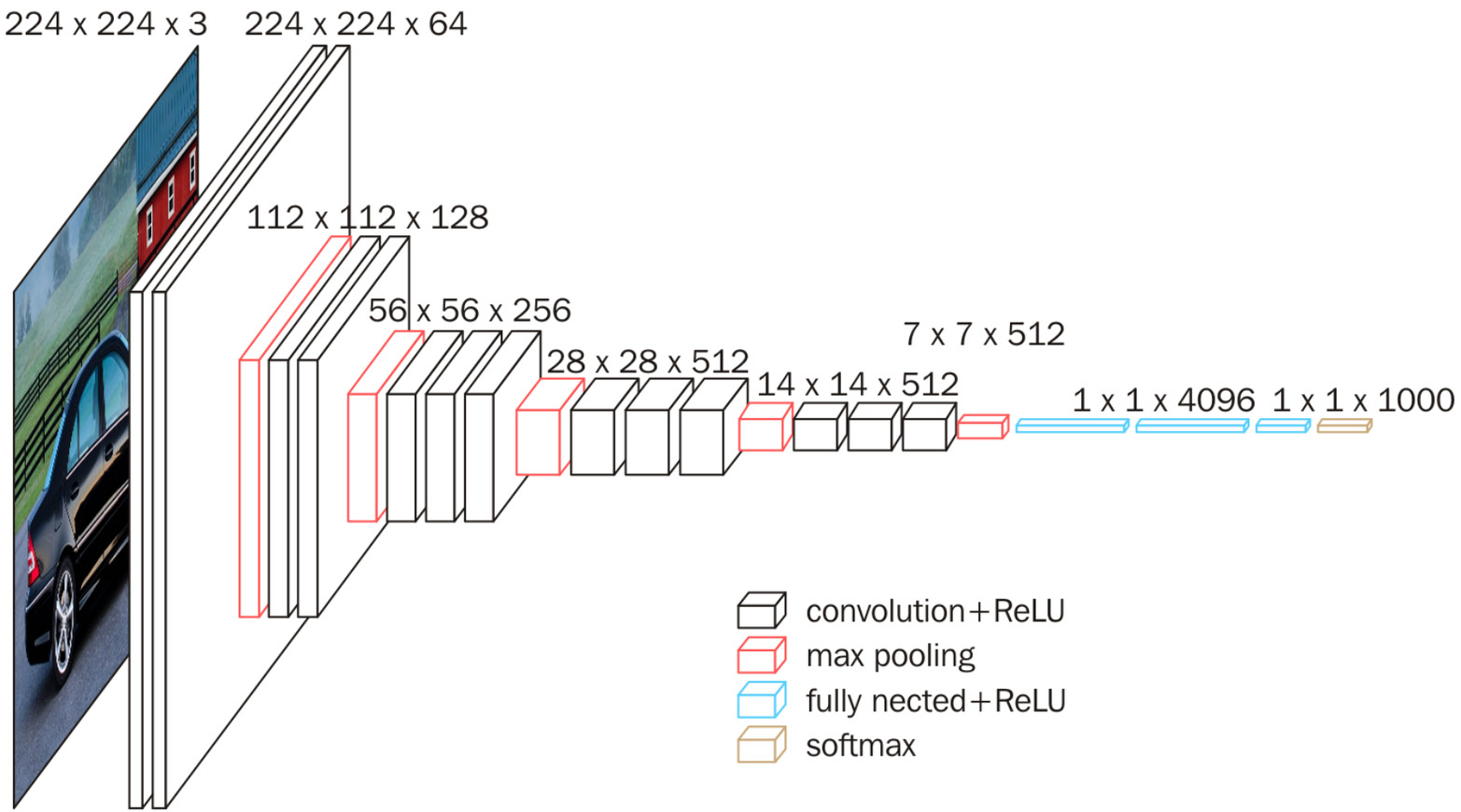
[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT for feature extraction

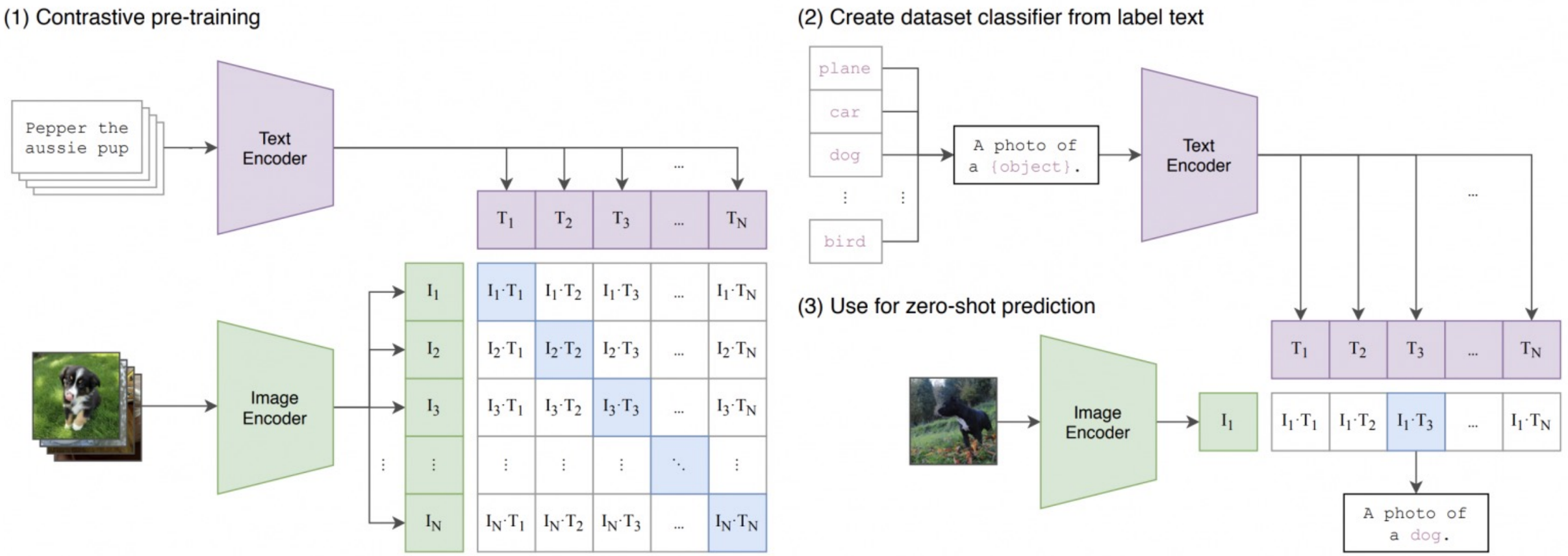


[1810.04805] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

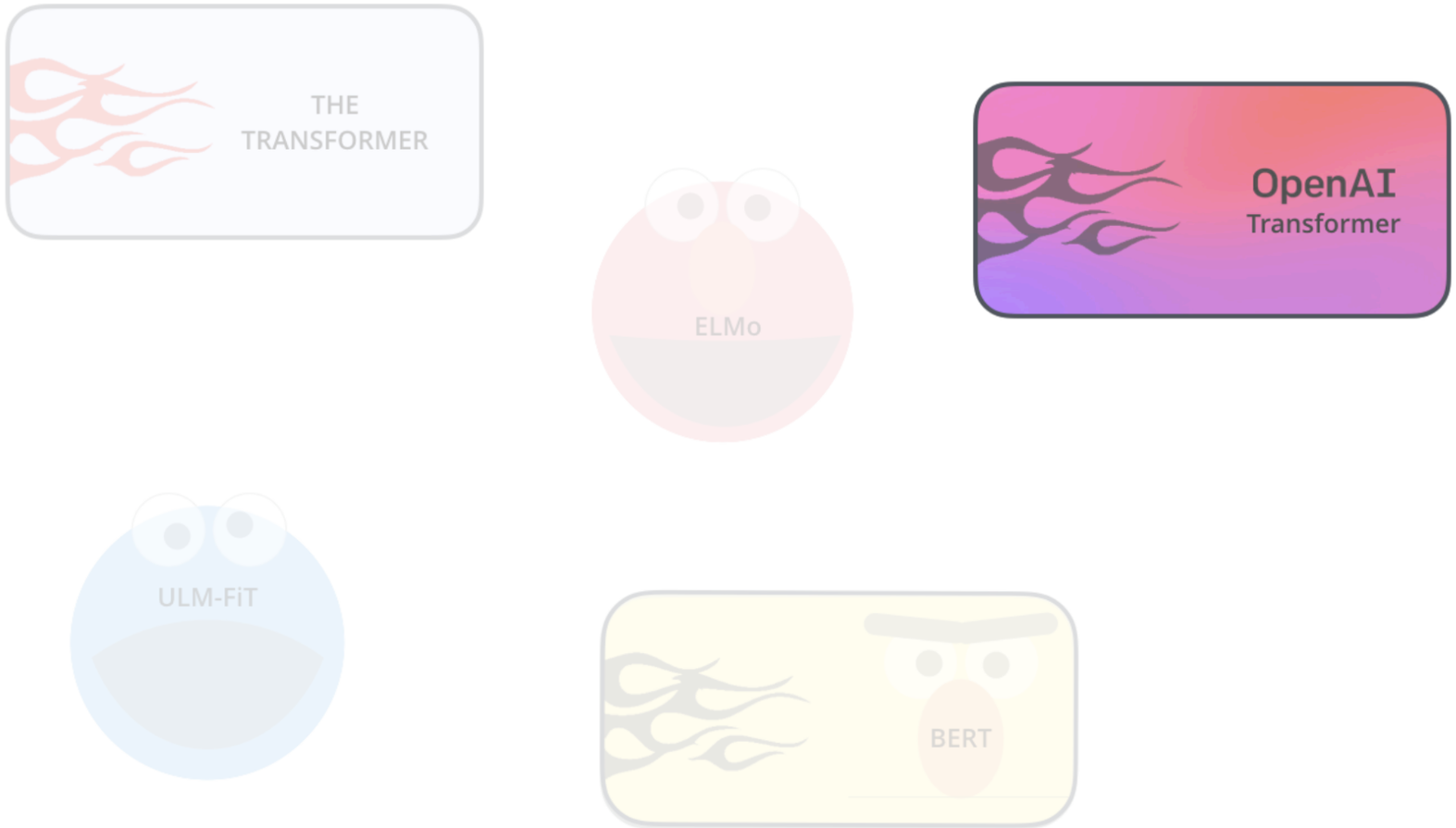
Recap: Applying transfer learning in CV



CLIP

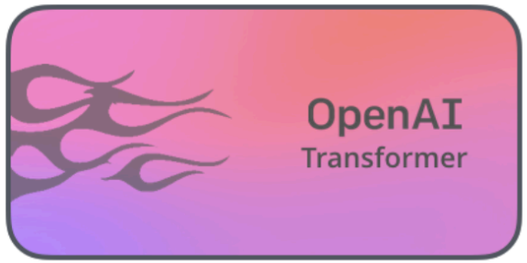

(1) Contrastive pre-training
Pepper the aussie pup
Text Encoder
Image Encoder
(2) Create dataset classifier from label text
plane
car
dog
bird
A photo of a {object}.
Text Encoder
(3) Use for zero-shot prediction
Image Encoder
A photo of a dog.

Другие модели

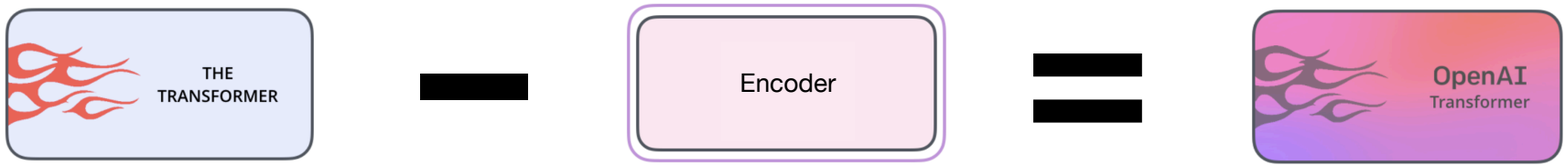


The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)

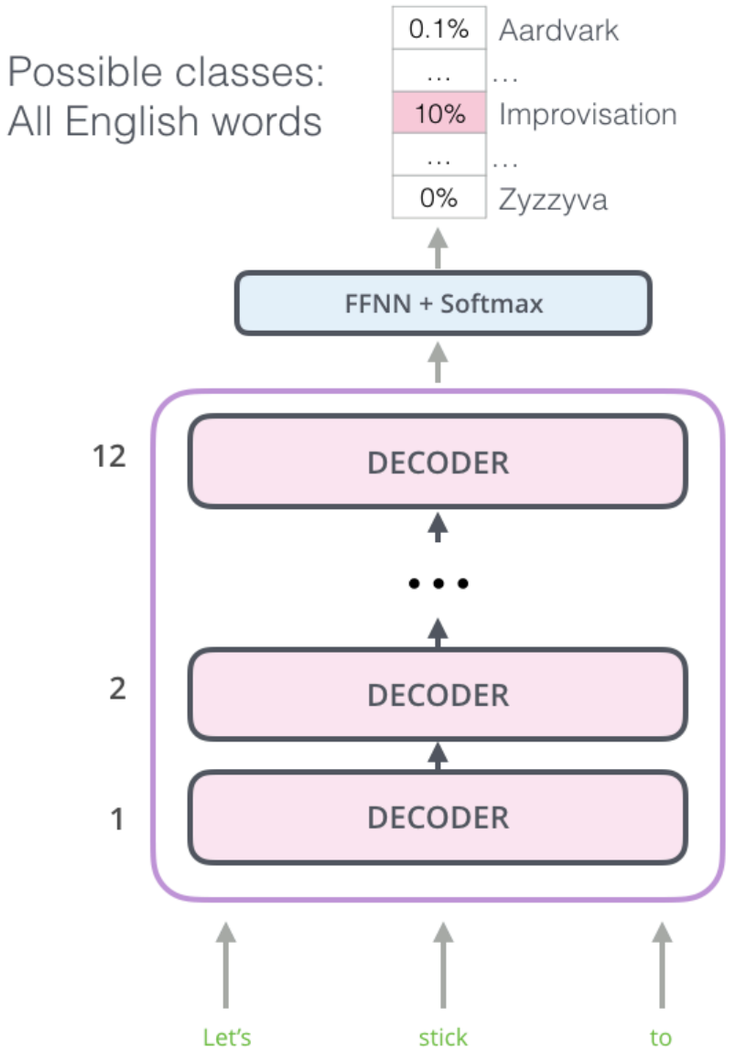
OpenAI Transformer



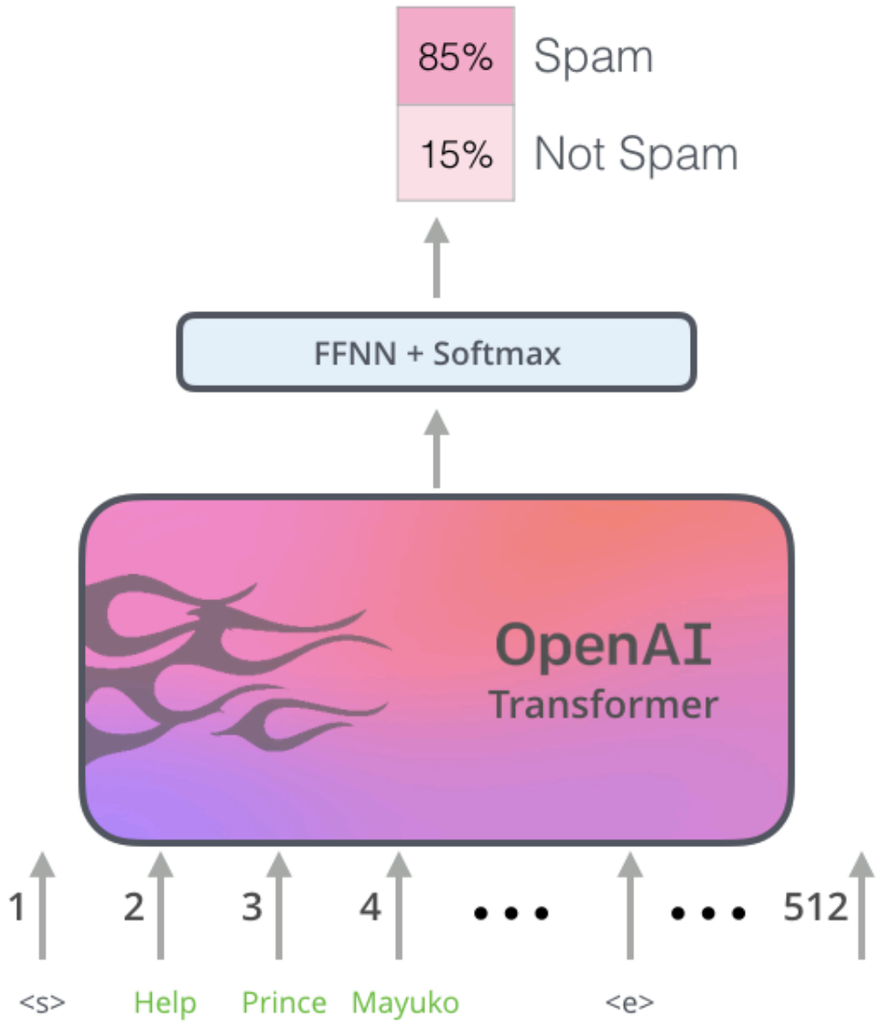
Pre-training a Transformer Decoder for Language Modeling



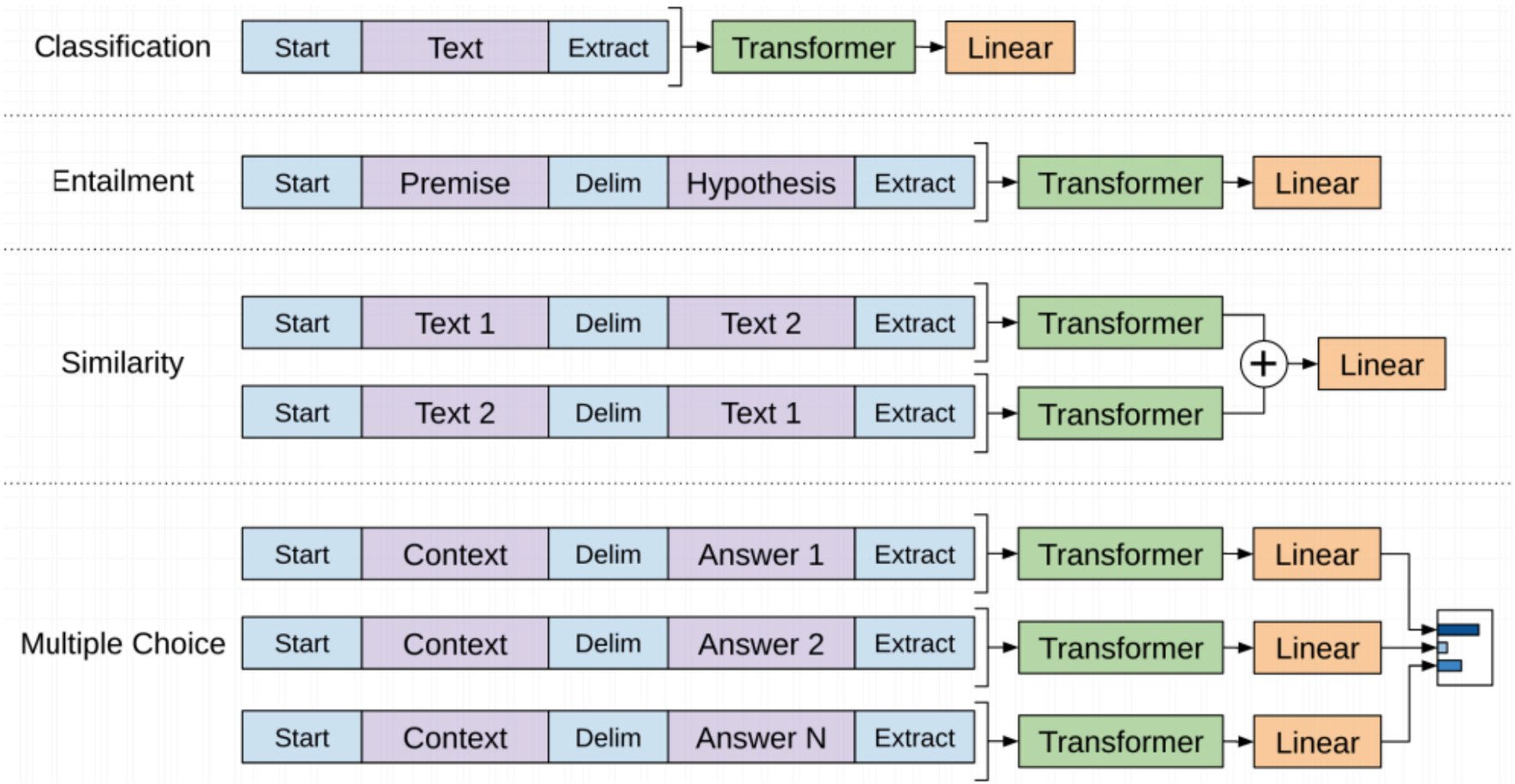
OpenAI Transformer



OpenAI Transformer



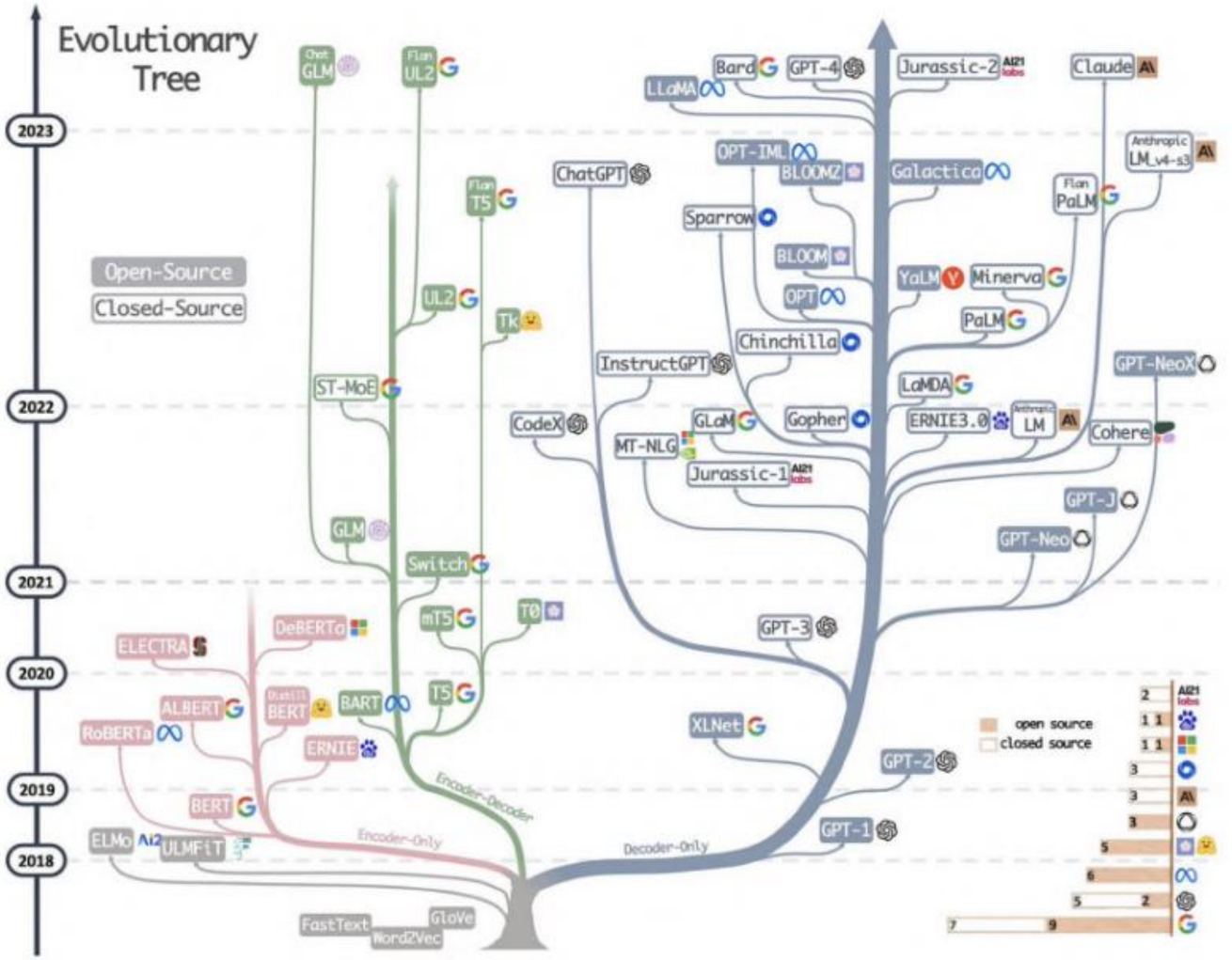
OpenAI Transformer



Сейчас «LLM» ≈ «AI»

Современные AI-сервисы:

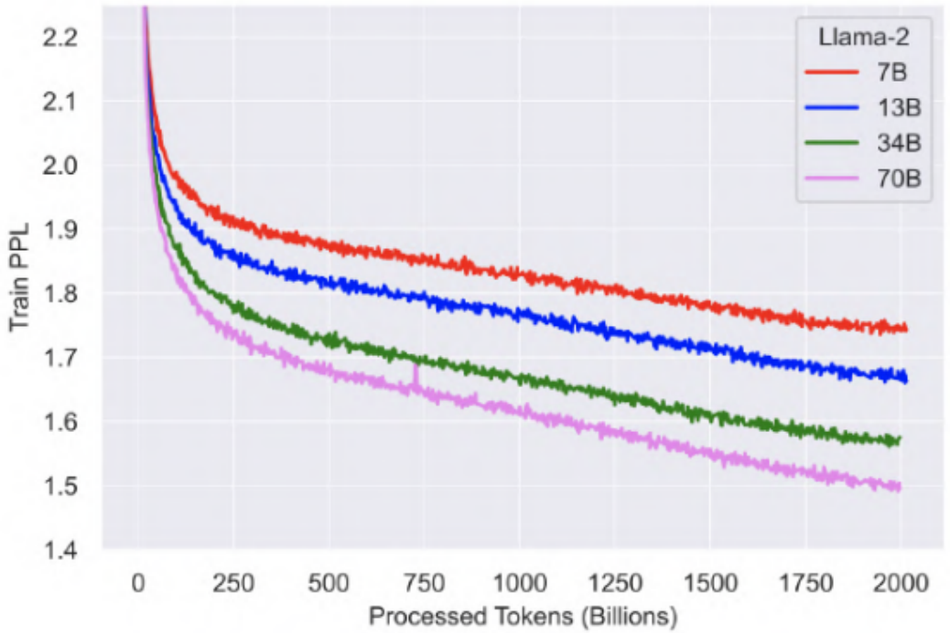
- **общаются** с людьми на любые темы на естественном языке
- **решают** школьные и университетские задачи по разным дисциплинам
- **понимают** и **генерируют** тексты, изображения, аудио



https://mellain.github.io/data/YNDX_Meetup_LLM_Train_Slides_2024.pdf

Этапы обучения

- Первая стадия обучения LLM — предобучение (pre-train)
- Модель учится предсказывать следующий токен по контексту слева
- Если учить с Teacher Forcing — контекст берётся из обучения, если без — из того, что сгенерировала в процессе сама модель (комбинируют)
- На этом этапе приобретает основные знания о языке и мире

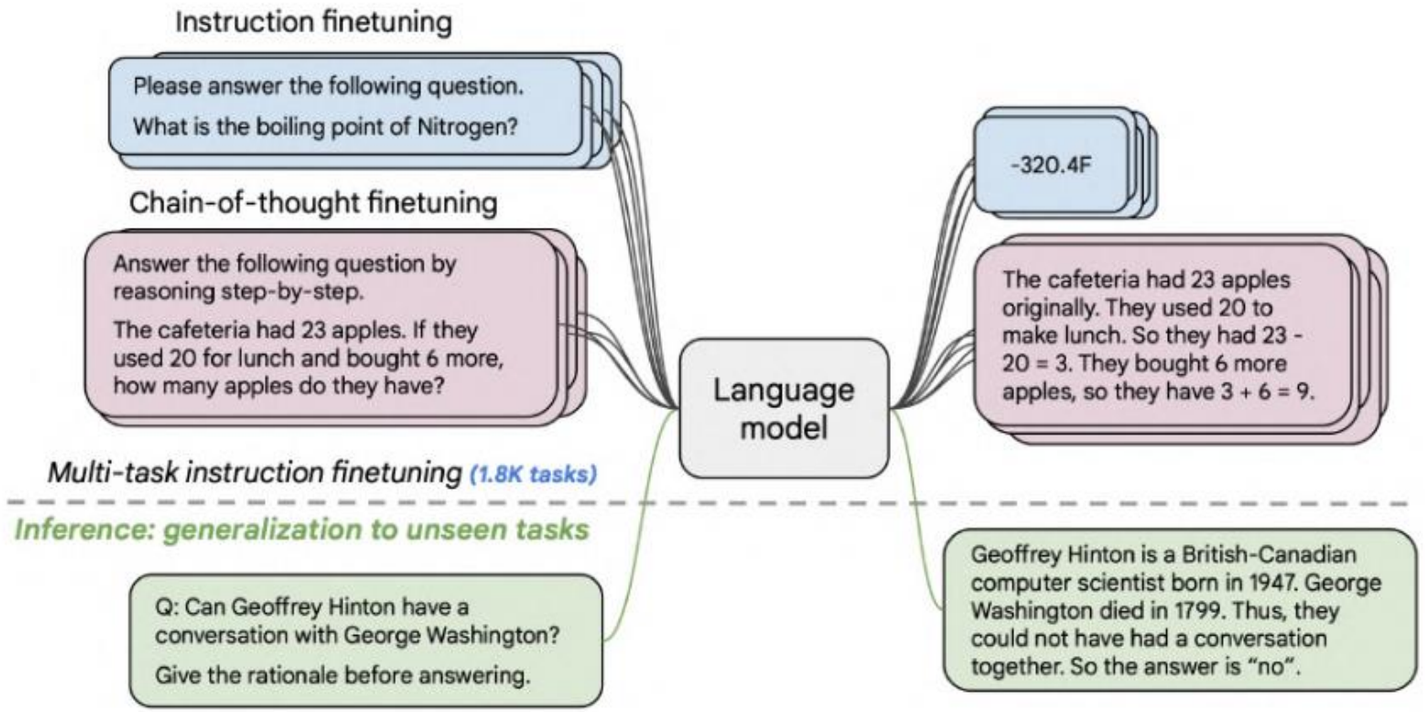


Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023

https://mellain.github.io/data/YNDX_Meetup_LLM_Train_Slides_2024.pdf

Этапы обучения

- Вторая стадия — Instruction Tuning (SFT)
- Модель учится понимать и исполнять запросы людей на естественном языке и вести диалоги
- Например: pre-train — LLaMA, instruct-tuned — Alpaca или Vicuna



Scaling Instruction-Finetuned Language Models, 2023

https://mellain.github.io/data/YNDX_Meetup_LLM_Train_Slides_2024.pdf

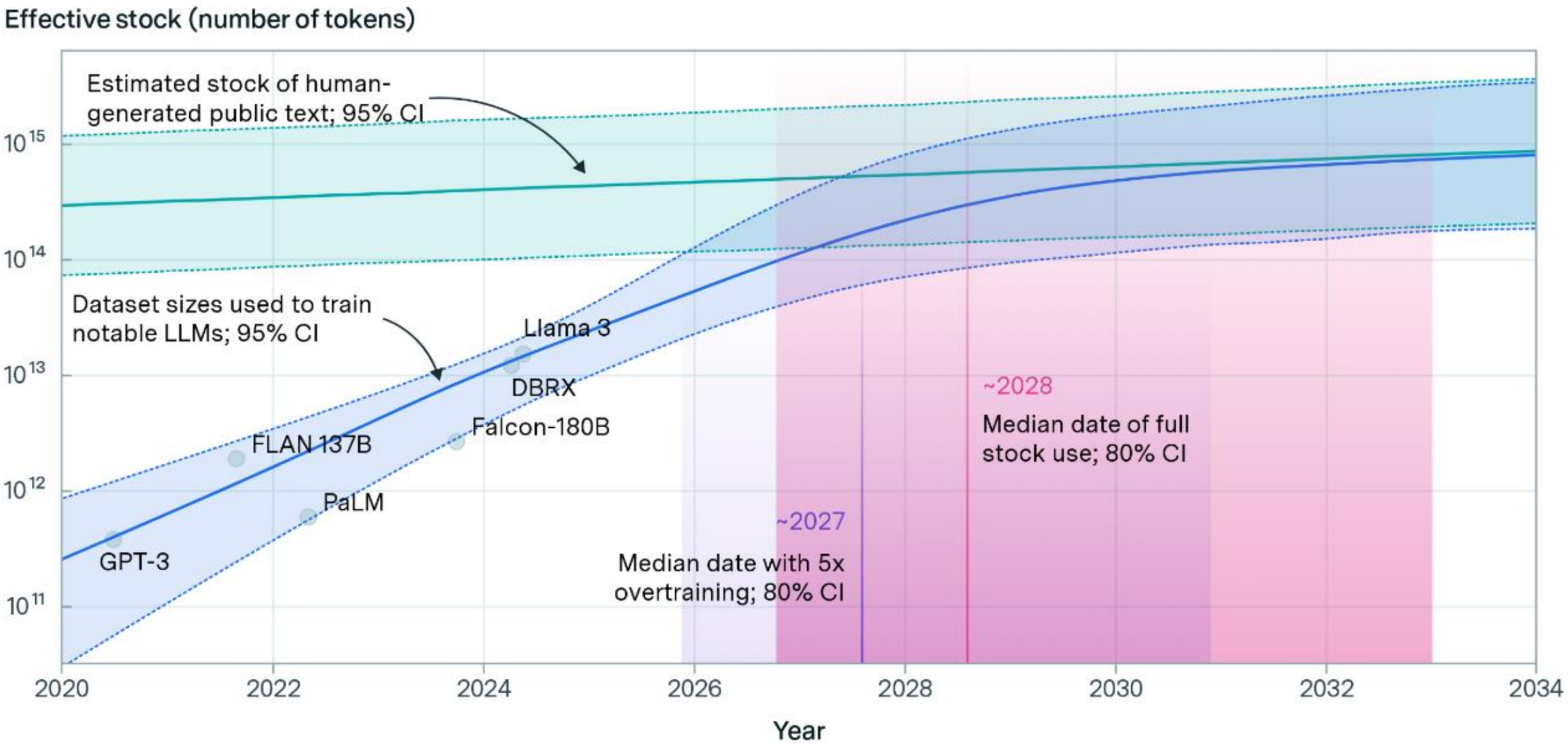
Этапы обучения

- Третий, опциональный, шаг — выравнивание (alignment)
- Диалоговая модель дообучается для генерации более корректных, полезных и безопасных ответов
- Популярная техника, использованная в Instruct GPT — RLHF:
  - обученная LLM генерирует на тестовом наборе инструкций ответы
  - ответы размечаются асессорами, на их ответах учится сильная reward-модель, она оценивает по тексту его качество
  - заводится две копии модели (A) и (B), учится (A)
  - обе модели генерируют ответы на каждый промпт, ответ (A) оценивается reward-моделью
  - веса (A) обновляются так, чтобы максимизировать reward и не давать ответы, очень далёкие от исходной (B)
  - расстояние определяется по KL-дивергенции между выходными распределениями моделей
  - обновление весов идёт по заданному алгоритме (PPO или A2C)

https://mellain.github.io/data/YNDX_Meetup_LLM_Train_Slides_2024.pdf

Projections of the stock of public text and data usage

EPOCH AI



https://epoch.ai/blog/will-we-run-out-of-data-limits-of-llm-scaling-based-on-human-generated-data