

Large Language Models (LLMs)

NLP: приложения

≡ Google Translate ⋮

Text Documents

ENGLISH - DETECTED RUSSIAN ENGLISH SPANISH RUSSIAN ENGLISH SPANISH

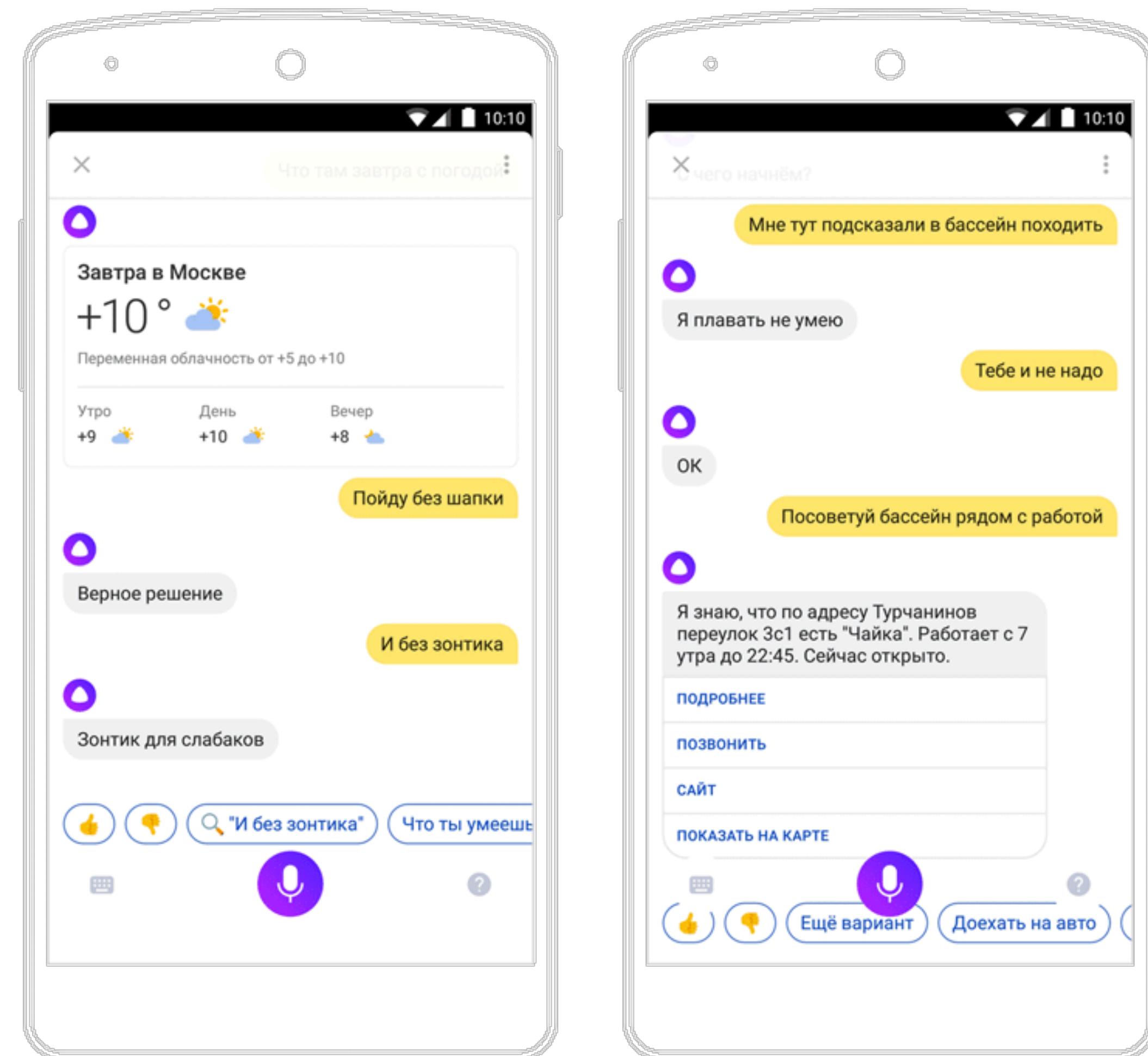
They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

Они были последними людьми, от которых вы ожидали быть вовлеченными во что-то странное или таинственное, потому что они просто не выдержали такой чепухи.

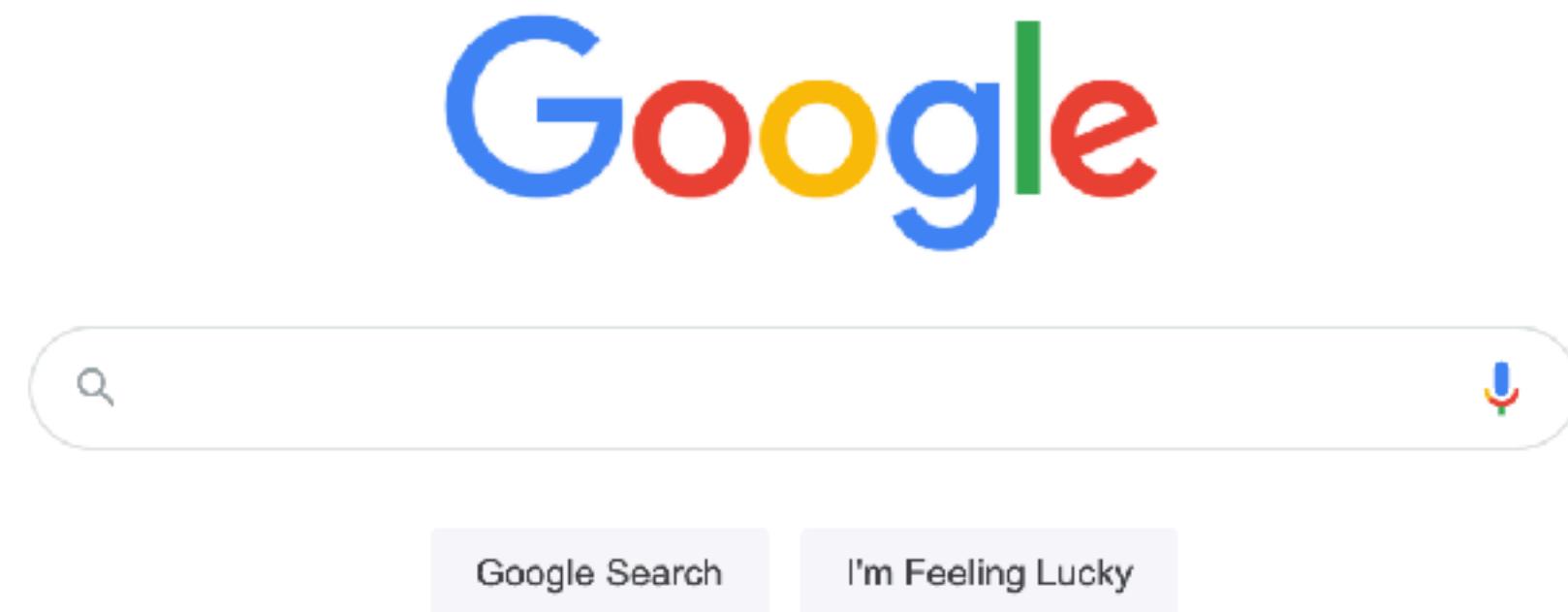
Oni byli poslednimi lyud'mi, ot kotorikh vy ozhidali byt' vovlechennymi vo chto-to strannoye ili tainstvennoye, potomu chto oni prosto ne vyderzhali takoy chepukhi.

Send feedback

NLP: приложения



NLP: приложения



Представления текста

Хотим: перевести слова в числовой вид

Представления текста

Хотим: перевести слова в числовой вид

One-hot encoding

Создаем словарь всех слов (размера V) и нумеруем их

Rome	=	[1, 0, 0, 0, 0, 0, ..., 0]
Paris	=	[0, 1, 0, 0, 0, 0, ..., 0]
Italy	=	[0, 0, 1, 0, 0, 0, ..., 0]
France	=	[0, 0, 0, 1, 0, 0, ..., 0]

Все вектора ортогональны друг другу

$$\cos_{\text{sim}}(w_1, w_2) = \frac{w_1^T w_2}{\|w_1\|^2 \|w_2\|^2} = 0$$

Ортогональные вектора — это такие вектора, угол между которыми равен 90 градусов, что значит, что их скалярное произведение равно нулю.

Косинусное сходство (cosine similarity) — это мера, используемая для вычисления схожести между двумя векторами в многомерном пространстве. Оно определяется как косинус угла между векторами.

Основное преимущество косинусного сходства в том, что оно позволяет сравнивать векторы, игнорируя их длину, и фокусируется исключительно на их направлениях. Неправильная интерпретация: One-hot encoding не учитывает семантические связи между словами (например, "кот" и "собака" рассматриваются как равные, хотя они имеют разное значение).

Представления текста

Хотим: перевести слова в числовой вид

Идея: вектора слов, которые встречаются в одном контексте, должны быть близки $\text{cos_sim}(w_1, w_2) \approx w_1^T w_2$

_____ *is the most beautiful capital city in Europe*

Rome? Paris?

Представления текста

Хотим: перевести слова в числовой вид

Идея: вектора слов, которые встречаются в одном контексте, должны быть близки $\cos_{\text{sim}}(w_1, w_2) \approx w_1^T w_2$

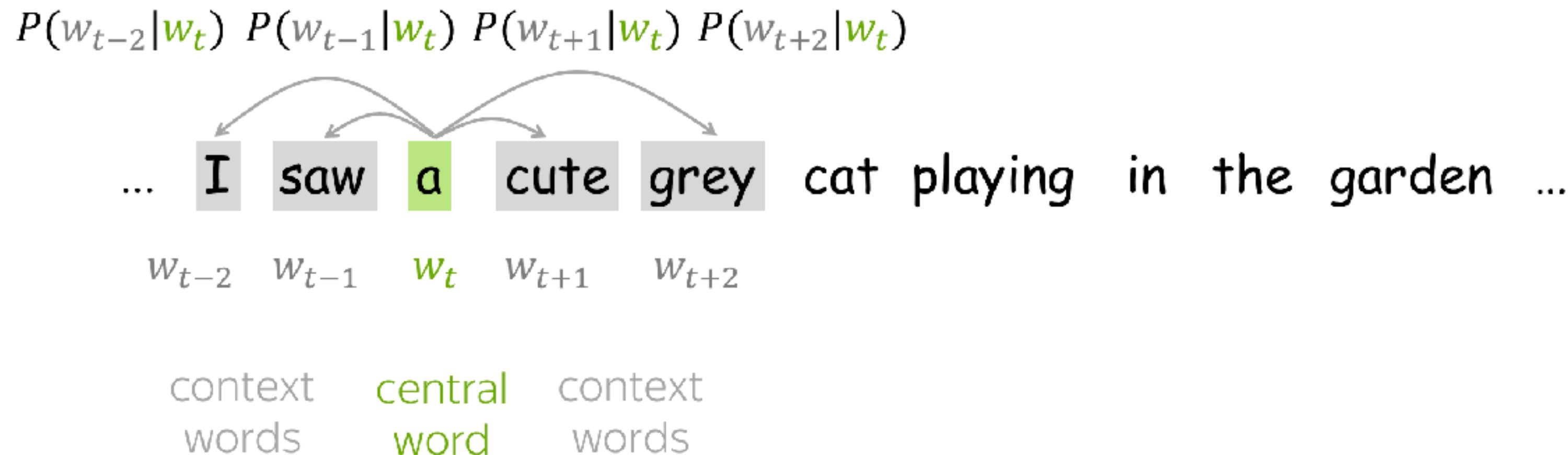
$$\mathbf{expect} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix}$$



Word2Vec

Хотим: перевести слова в числовой вид

Будем проходить **окном** по **большому корпусу текстов** и вычислять вероятность **центрального** слова при условии **контекста**

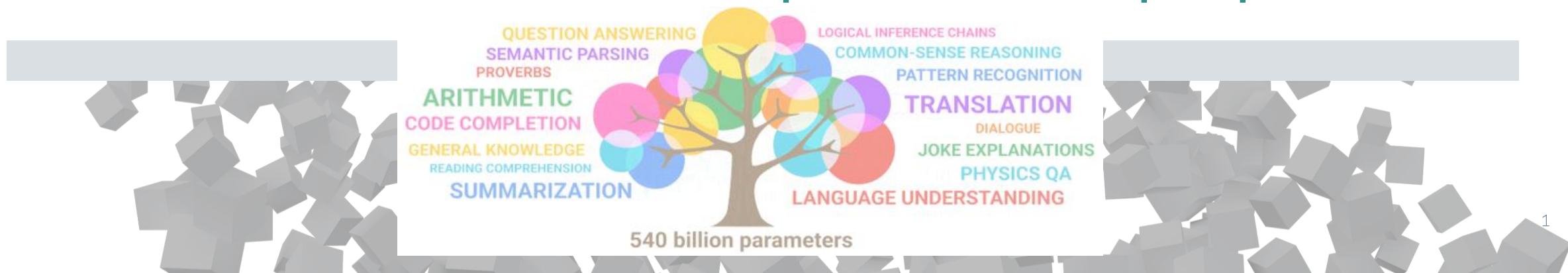


[Image credit](#)

История развития LLM

Эволюция нейросетевых языковых моделей					
2010-е	2018	2019	2020	01/2022	11/2022
Автозаполнение в смартфонах	GPT-1	GPT-2	GPT-3	GPT-3.5	ChatGPT
КЛЮЧЕВОЕ НОВОВВЕДЕНИЕ					
Первое массовое применение простейшей предсказательной модели	Использование архитектуры трансформера с большой масштабируемостью	В 10 раз больше параметров модели и тренировочных данных	Еще в 100 раз больше параметров и в 10 раз – тренировочных данных	Дообучение на обратной связи от живых людей	Удобный интерфейс в формате чата и бесплатный доступ для широких масс
ЭФФЕКТ					
Телефон догадывается, что человек хотел сказать по первым вводимым символам	Подтвердилаась возможность использовать трансформер для генерации текста	Научились выдавать связные тексты, зачастую вполне похожие на человеческие	Внезапно появилось много новых навыков: арифметика, рассуждение, кодинг	Научились подгонять ответы так, чтобы они нравились людям	Большая популярность и толчок к коммерческому применению

Изменение “способностей” LLM с ростом количества параметров

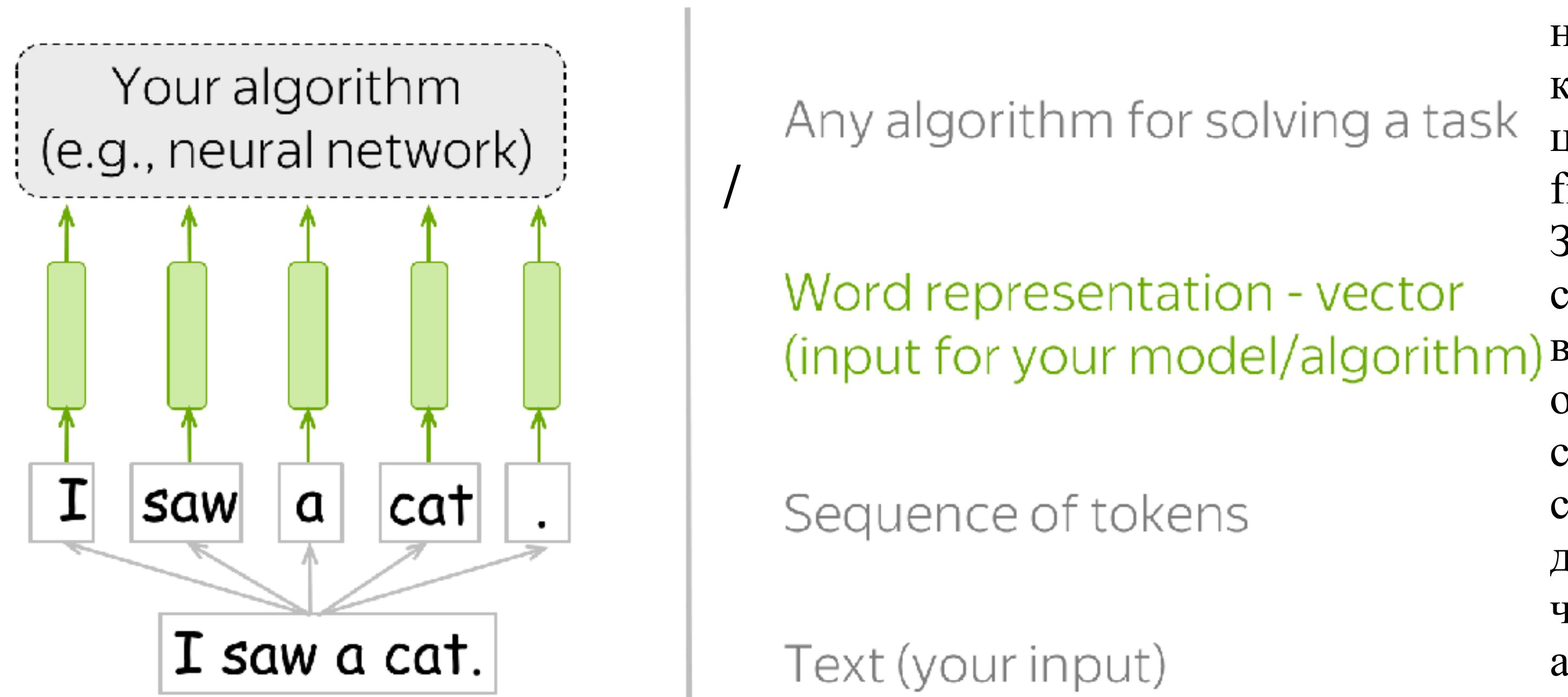


Pre-trained Transformer

Основная идея: взять предобученные эмбеддинги и дообучить на нужную задачу

Pre-train + Fine-tune (Transfer Learning)

Transfer learning (перенос обучения) — это подход в машинном обучении, при котором модель, обученная на одной задаче, используется для решения другой, связанной задачи.

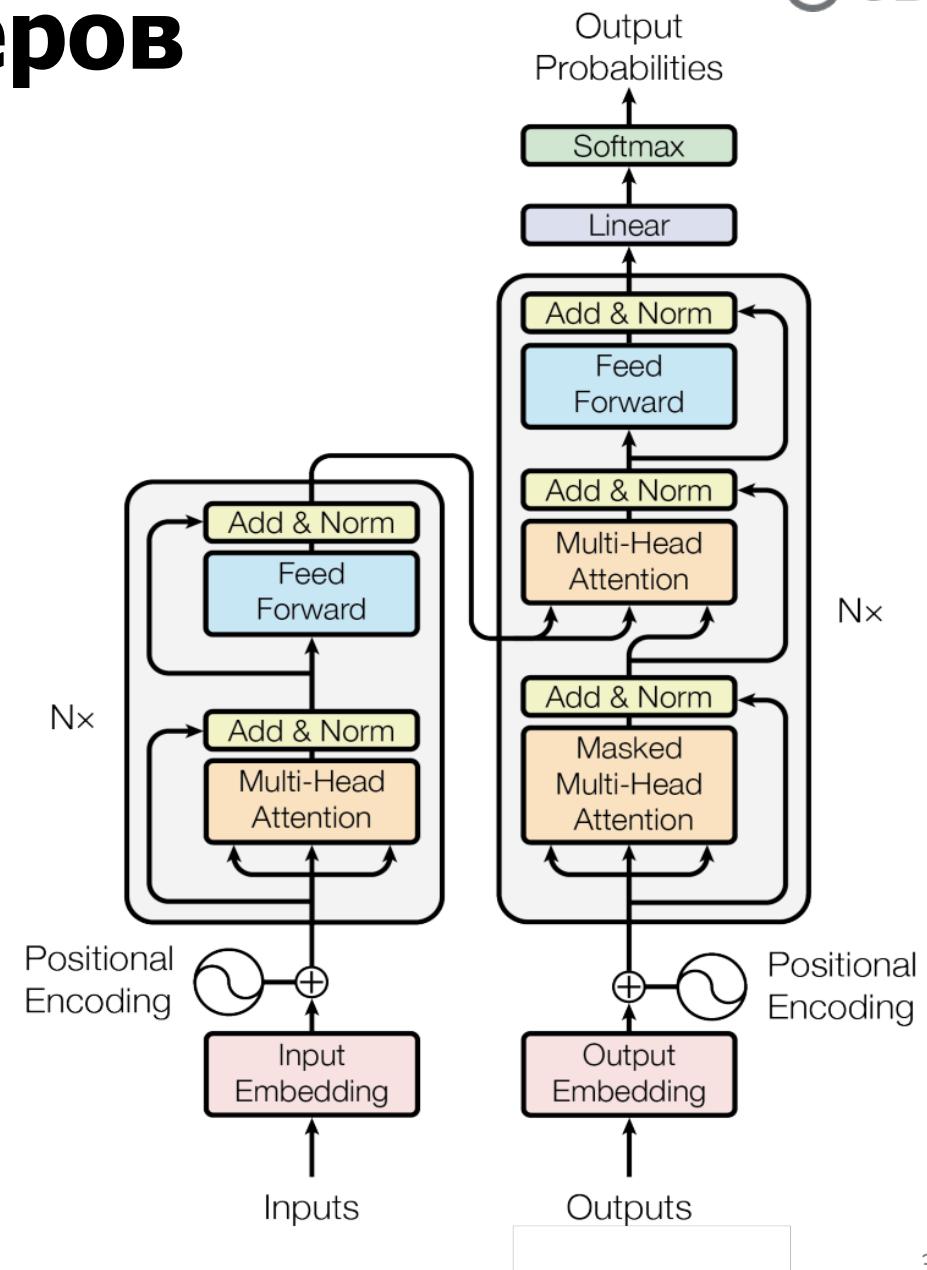


Fine-tuning (дообучение) — это процесс адаптации предварительно обученной модели к новой, часто более специфичной задаче или набору данных, разновидность transfer learning. Как работает: получаем предобученную модель, адаптируем (fine-tune) с помощью небольшого набора данных, который имеет отношение к целевой задаче. В процессе fine-tuning можно: Замораживать некоторые слои модели, оставляя их веса фиксированными, и обучать только верхние слои, которые отвечают за специфику новой задачи, дообучать все слои модели, чтобы она могла адаптировать свои представления под специфическую задачу. Это требует больше данных и вычислительных ресурсов. [Image credit](#)

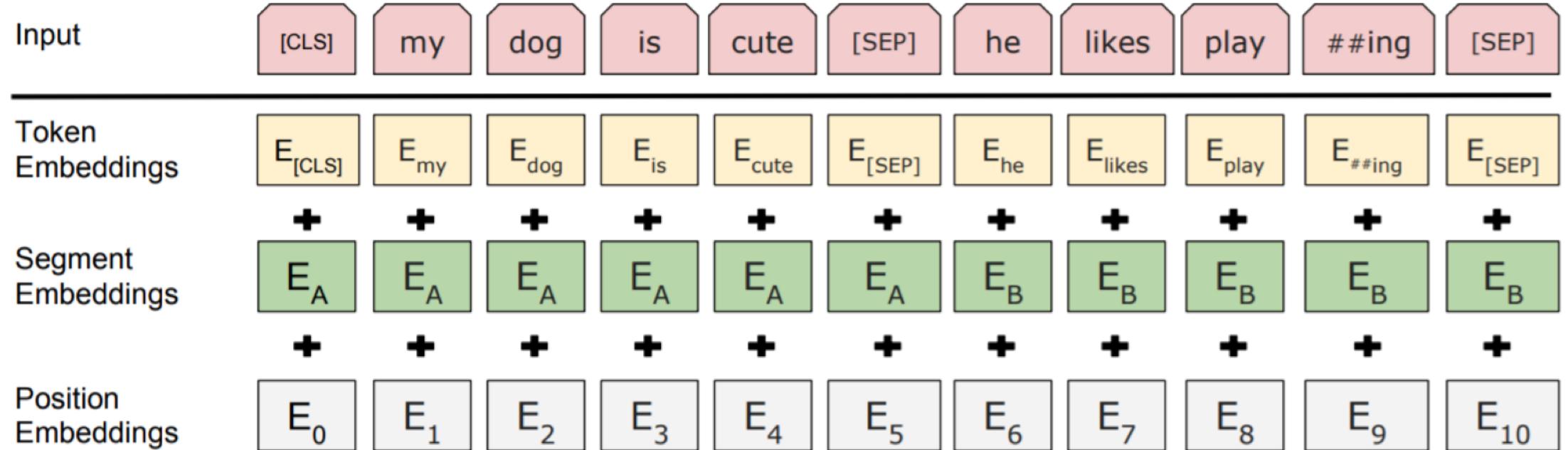
Архитектура трансформеров

Архитектура состоит из двух типов кодировщиков: энкодеры – левая часть трансформера и декодеры – правая половина трансформера, которые состоят из повторяющихся блоков, представляющих собой последовательность:

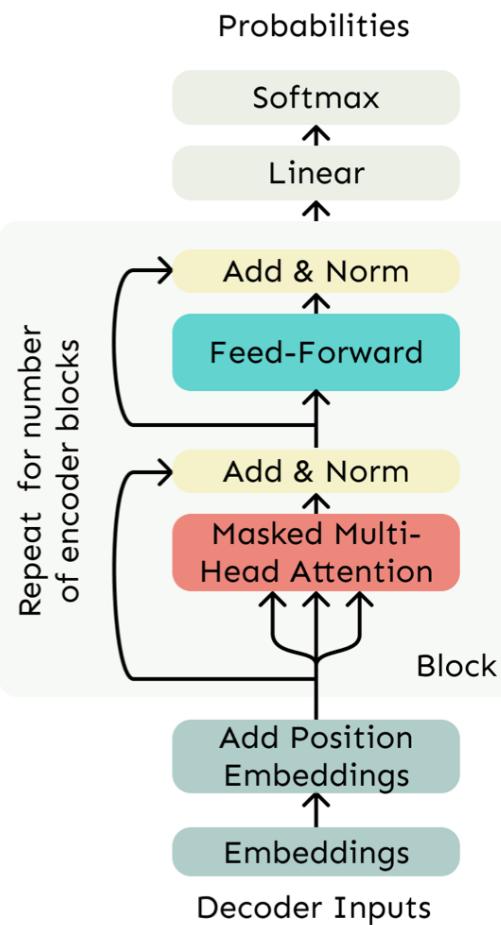
- Self-attention
- Суммирование выходов и нормализация слоя
- Residual connections
- Полносвязные слои нейронной сети



Embeddings + Positioning

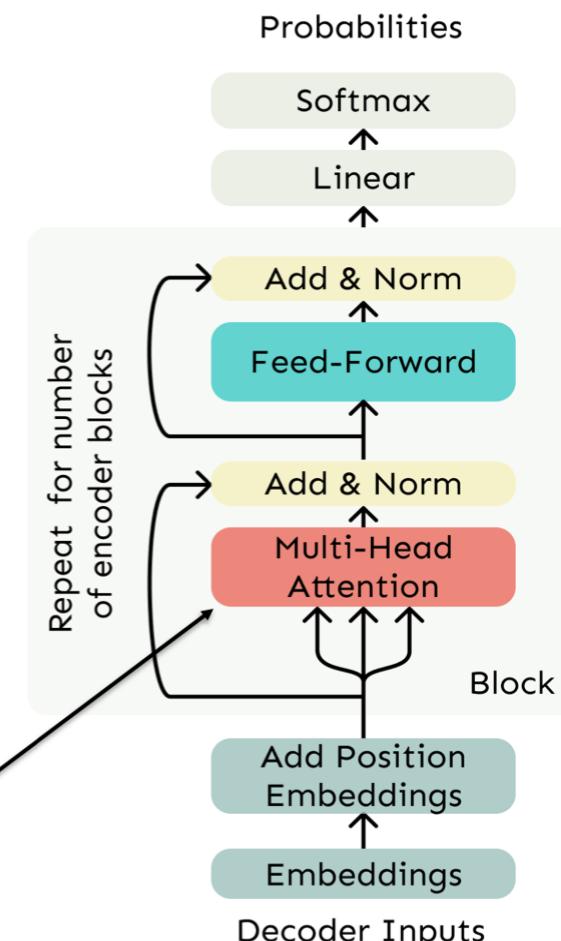


Decoder



- Блоки:
 - Self-attention
 - Add & Norm
 - Feed-Forward
 - Add & Norm
- Decoder ограничивается односторонним контекстом
- Encoder позволяет учитывать контекст по обе стороны от слова, для этого в self-attention убирается маскирование

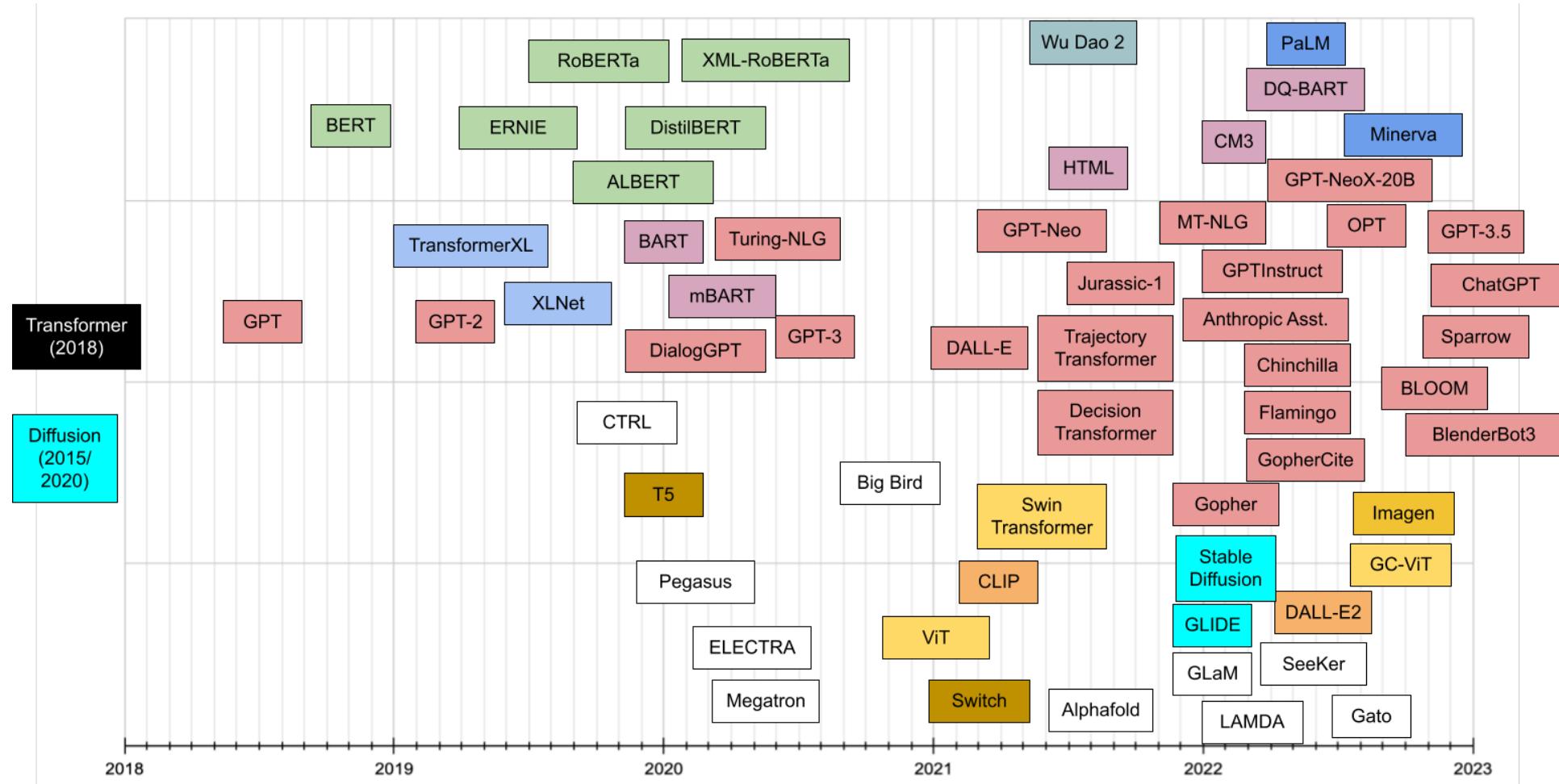
Encoder



Хронология развития трансформеров

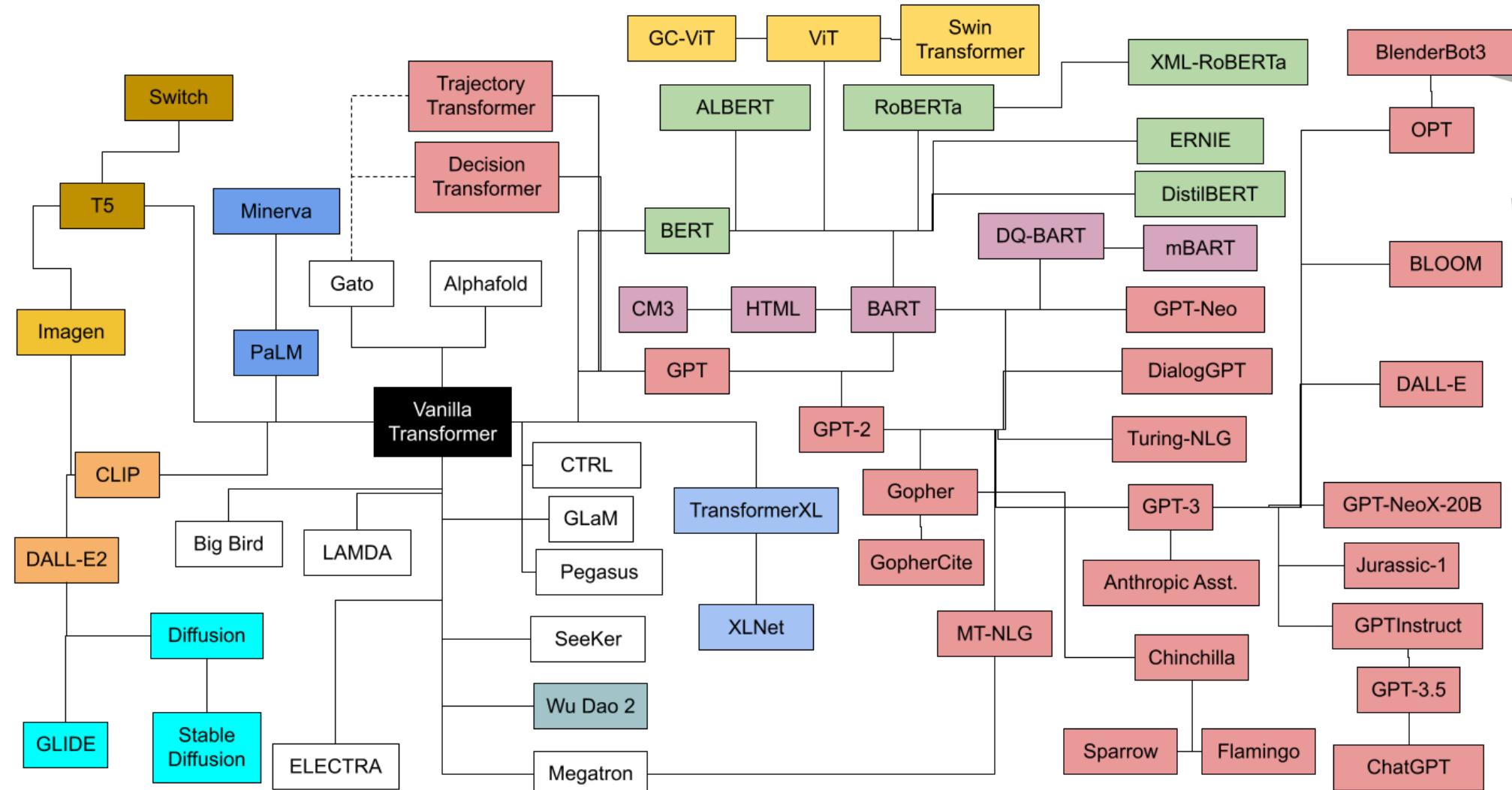
<https://arxiv.org/pdf/2303.18223.pdf>

<https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/>



Дерево связей

<https://amatriain.net/blog/transformer-models-an-introduction-and-catalog-2d1e9039f376/>

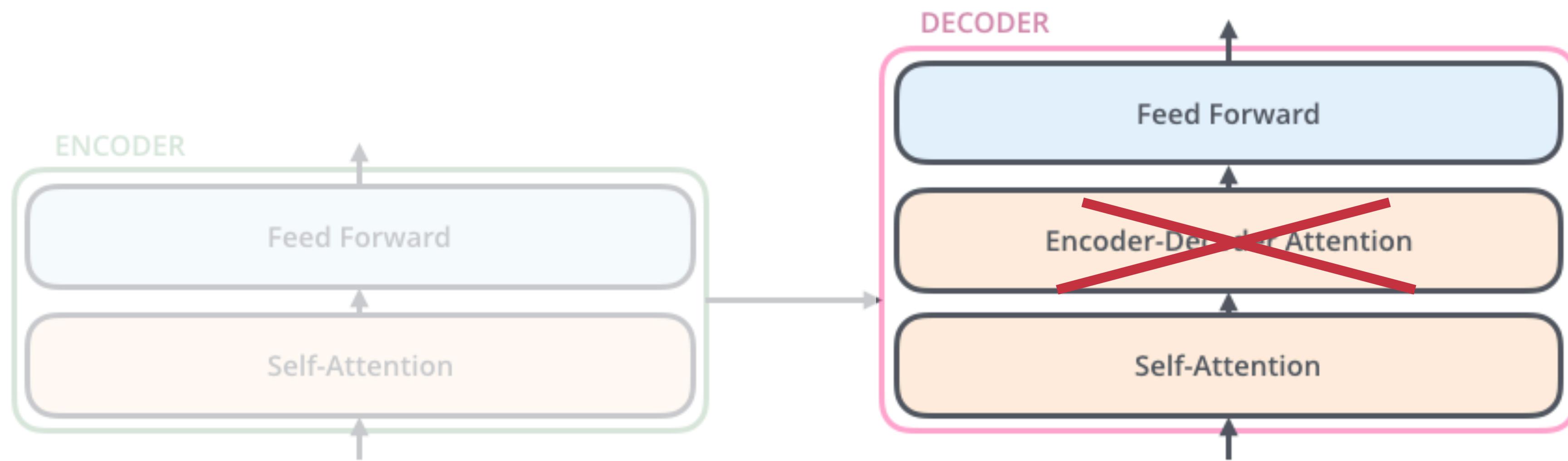


GPT

GPT

Generative Pre-Training

Архитектура: Transformer Decoder



[Image credit](#)

GPT

Generative Pre-Training

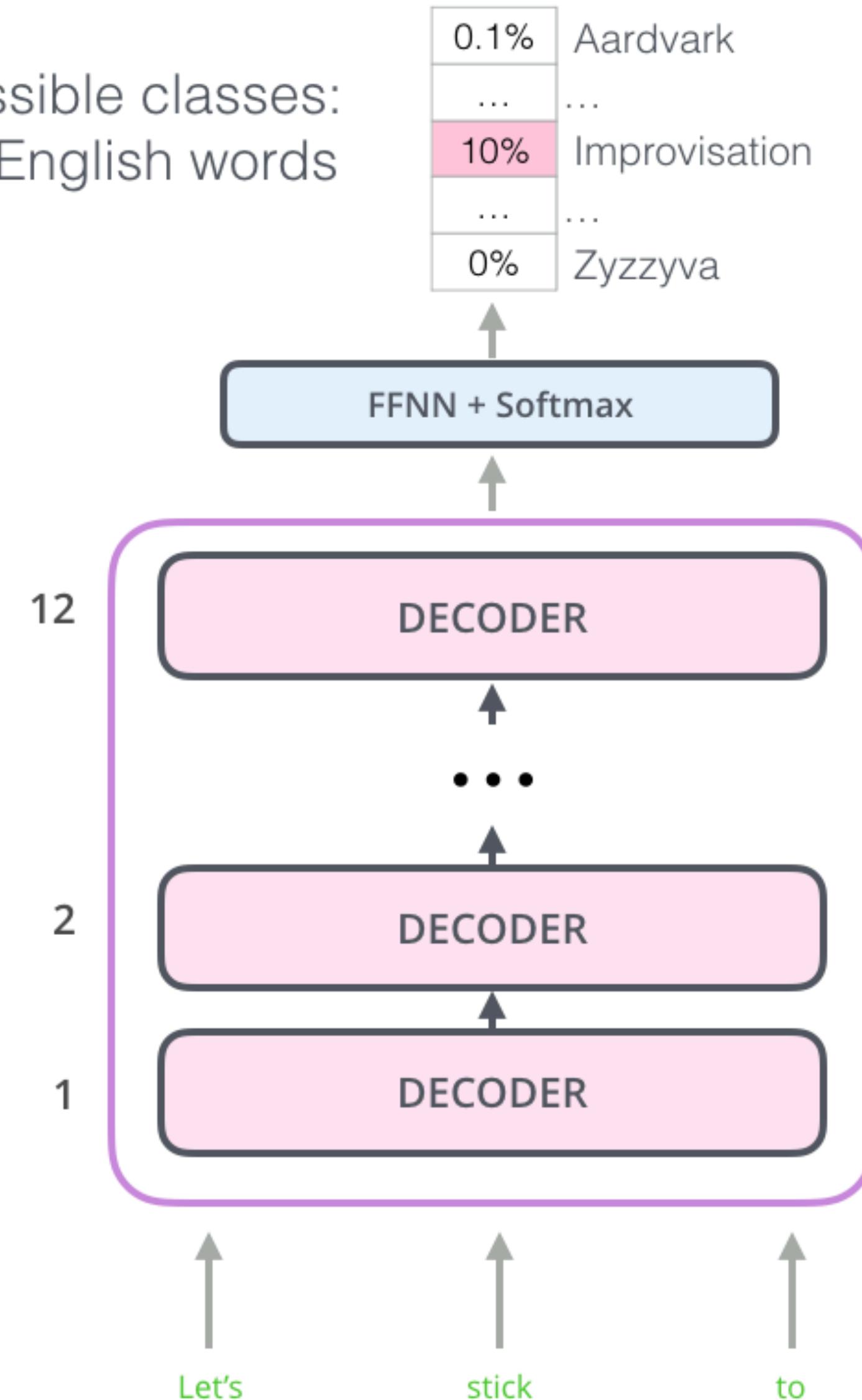
Архитектура: Transformer Decoder

Данные: тексты книг (BookCorpus)

- + Разнообразие
- + Большой объем

Possible classes:
All English words

0.1%	Aardvark
...	...
10%	Improvisation
...	...
0%	Zyzyva



GPT

Generative Pre-Training

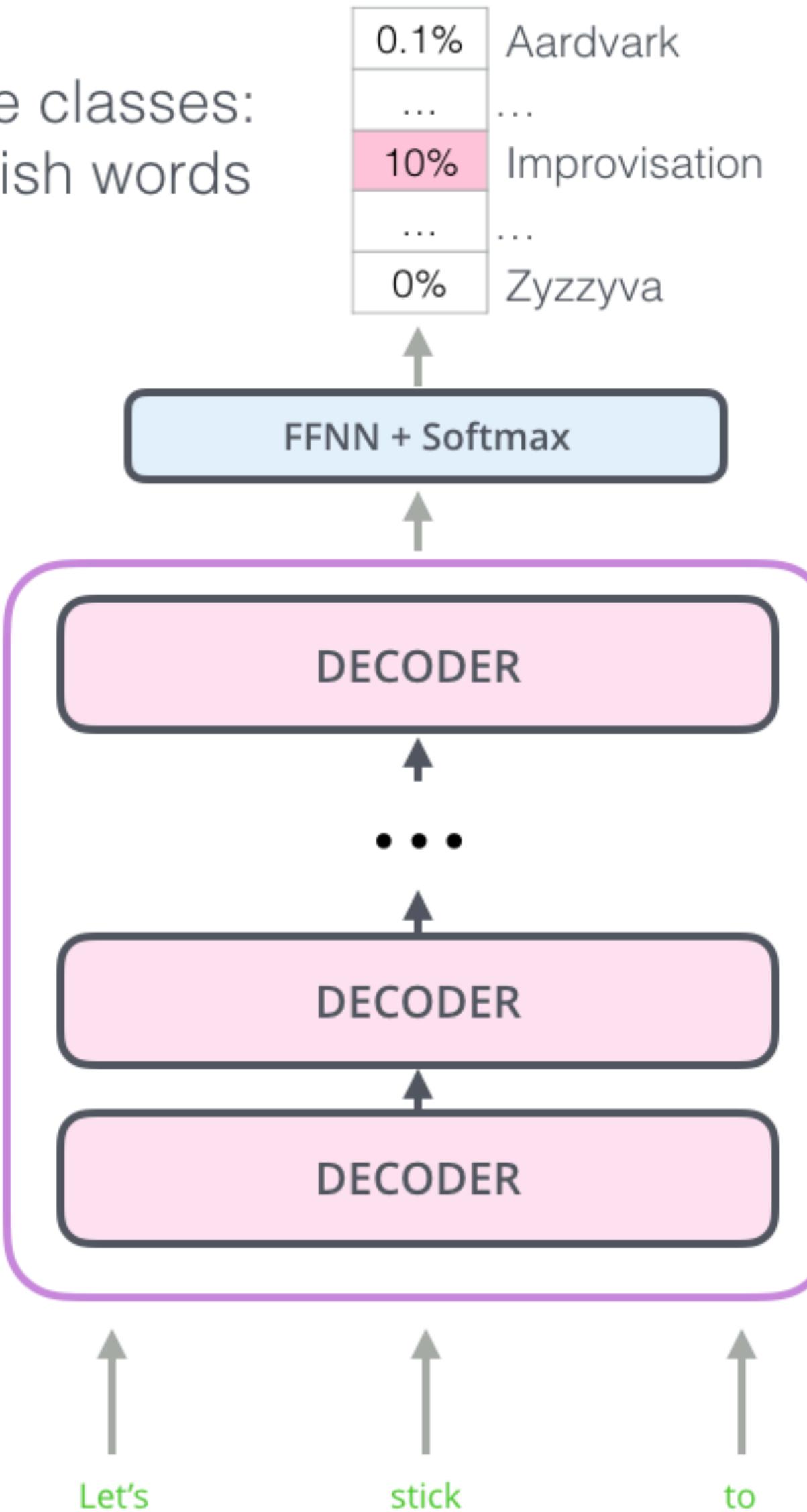
Архитектура: Transformer Decoder

Данные: тексты книг (BookCorpus)

Задача для обучения: Language Modeling

Possible classes:
All English words

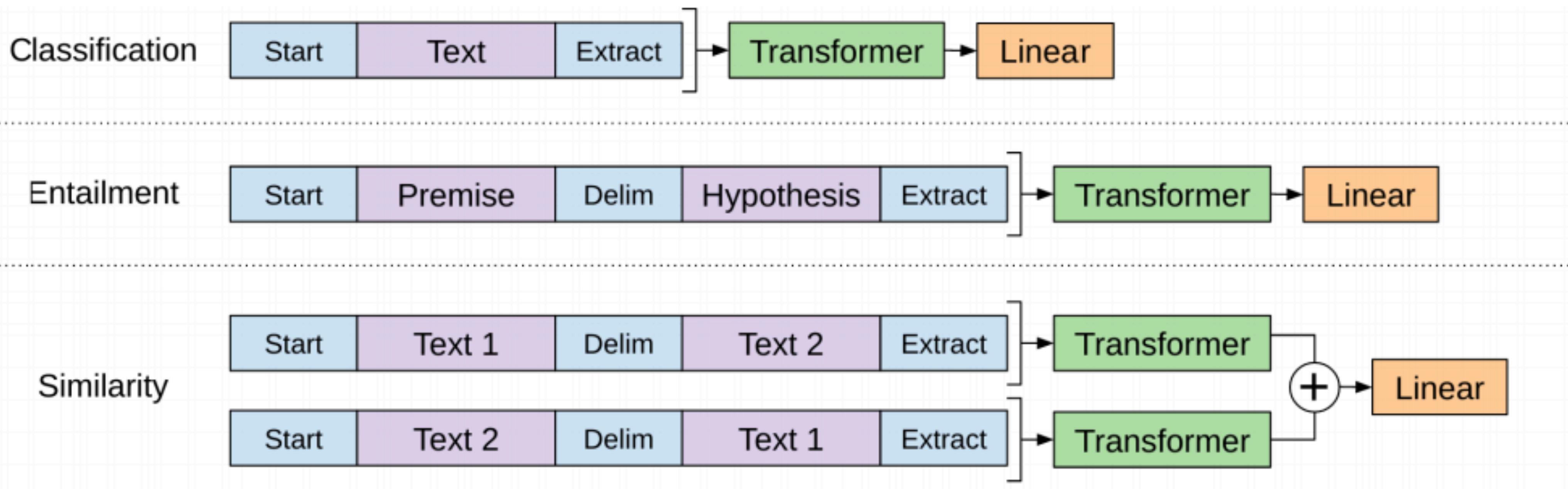
0.1%	Aardvark
...	...
10%	Improvisation
...	...
0%	Zyzyva



GPT

Как использовать?

Для разных задач - разный формат входа



GPT-2

х10 параметров

х10 данных

GPT-2

Как использовать? **zero-shot task transfer**

Нет дообучения на новую задачу (fine-tuning)

Форматируем input, чтобы была понятна задача

Zero-shot task transfer — это подход в машинном обучении, при котором модель обучена выполнять задачу без явного обучения на примерах этой задачи

В рамках zero-shot подхода вы можете просто предоставить модели описание задачи, например: "Классифицируй следующий текст в одну из категорий: политика, спорт, наука, искусство." И, например, передать сам текст: "Сегодня прошел важный матч в Лиге чемпионов, где победу одержал клуб из Барселоны."

GPT-2

Форматируем input, чтобы была понятна задача

Пример: задача суммаризации

Article: Amina Ali Qassim is sitting with her youngest grandchild on her lap, wiping away tears with her headscarf. Only a few months old, this is the baby girl whose ears she desperately tried to cover the night the aerial bombardment started. She lay awake, she says, in a village mosque on the Yemeni island of Birim, counting explosions as the baby cried.

It could have been worse though. They could have still been in their house when the first missile landed.
"Our neighbor shouted to my husband 'you have to leave, they're coming.' And we just ran. As soon as we left the house, the first missile fell right by it and then a second on it. It burned everything to the ground," Qassim tells us
...

Input

TL;DR: Yemen is in the middle of a civil war. Saudi Arabia is leading the coalition bombing campaign. It's been bombing Yemen for more than two months now.

GPT-2 prediction

GPT-2

Language Modeling (генерация текста)

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.



GPT-3

x100 параметров (по сравнению с GPT-2)

x5 данных (по сравнению с GPT-2)

* нет в свободном доступе

GPT-3

х100 параметров (по сравнению с GPT-2)

х5 данных (по сравнению с GPT-2)

Как использовать? **zero-shot/one-shot/few-shot settings**

- показываем 0/1/несколько примеров

One-shot learning (обучение на одном примере) — это подход, при котором модель учится выполнять задачу, используя только один пример. Задача: Классификация текста. Пример 1: "Этот текст о футболе." Задача: "Определите, к какой категории относится следующий текст." Текст: "Сегодня прошел важный матч Лиги чемпионов." Ответ модели: "спорт".

В few-shot learning модель получает несколько примеров и на основе них обучается выполнять задачу. Этот подход использует методы обобщения и передачу знаний (transfer learning), чтобы эффективно работать с ограниченным количеством данных. Пример: Задача: Классификация текста. Примеры: "Текст о футболе." → категория: "спорт" "Текст о политике." → категория: "политика" Задача: "Классифицируйте следующий текст." Текст: "Сегодня прошел важный матч Лиги чемпионов." Ответ модели: "спорт"

GPT-3

x100 параметров (по сравнению с GPT-2)

x5 данных (по сравнению с GPT-2)

Как использовать? **zero-shot/one-shot/few-shot settings**

- показываем 0/1/несколько примеров

English sentence 1 = France sentence 1
English sentence 2 = France sentence 2
English sentence 3 =



GPT-3 prediction

GPT-3

Prompt:

The following are ideas for startups, apps, marketplaces, platforms:

###

A startup that helps content creators, such as photographers and videographers, monetize their work.

###

An online forum that helps people find roommates

###

GPT-3:

A community hub for entrepreneurs to share ideas and ask questions of experienced entrepreneurs

###

A website that helps people discover new interests and hobbies.

###

A company that helps brands advertise their products to potential buyers on Instagram.

###

Image credit

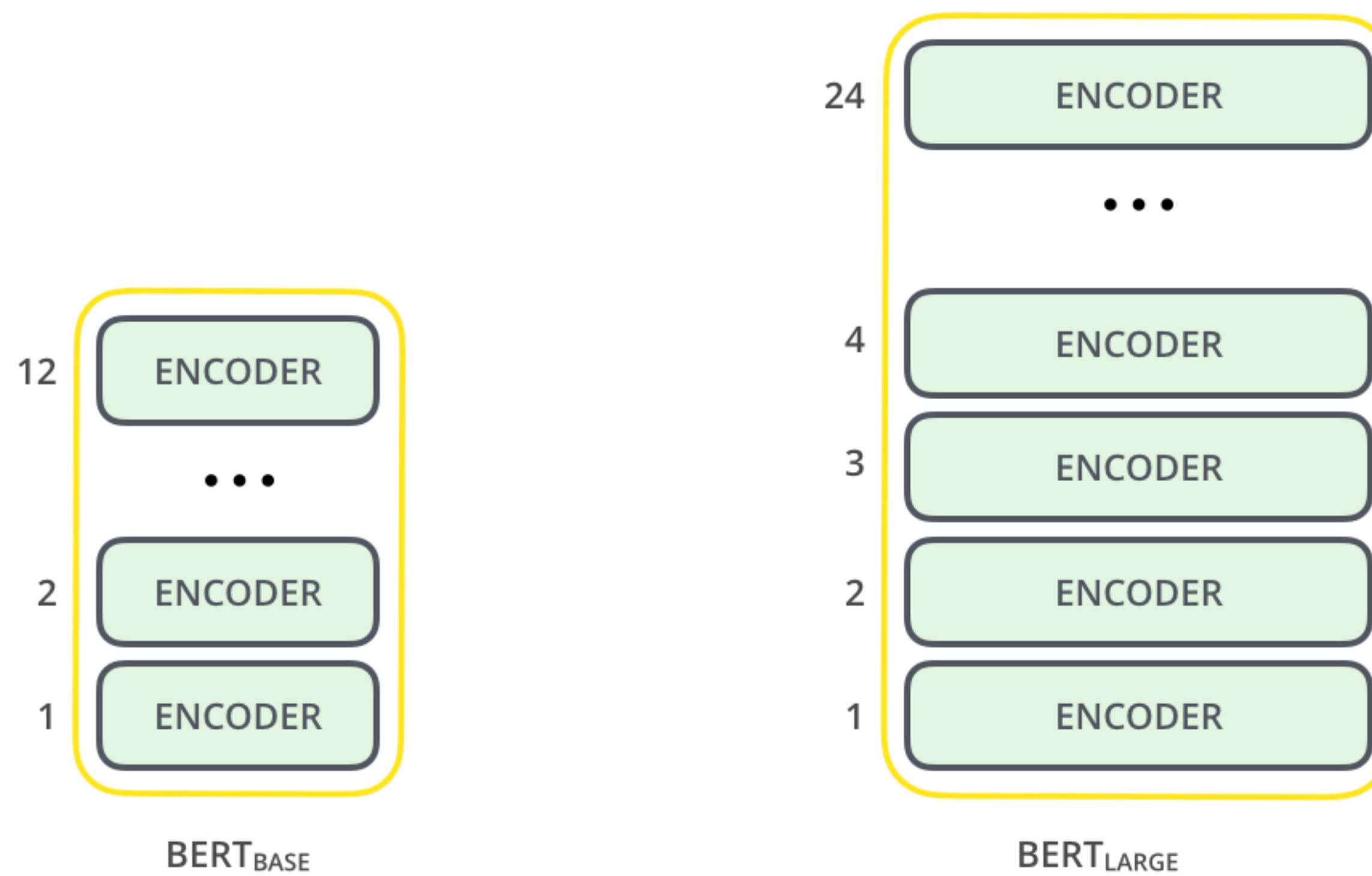
BERT



BERT

Bidirectional Encoder Representations from Transformers

Архитектура: Transformer Encoder



BERT

Bidirectional Encoder Representations from Transformers

Архитектура: Transformer Encoder

Данные: Wikipedia и тексты книг (BookCorpus)

Задачи для обучения: Masked LM и Next Sentence Prediction

BERT

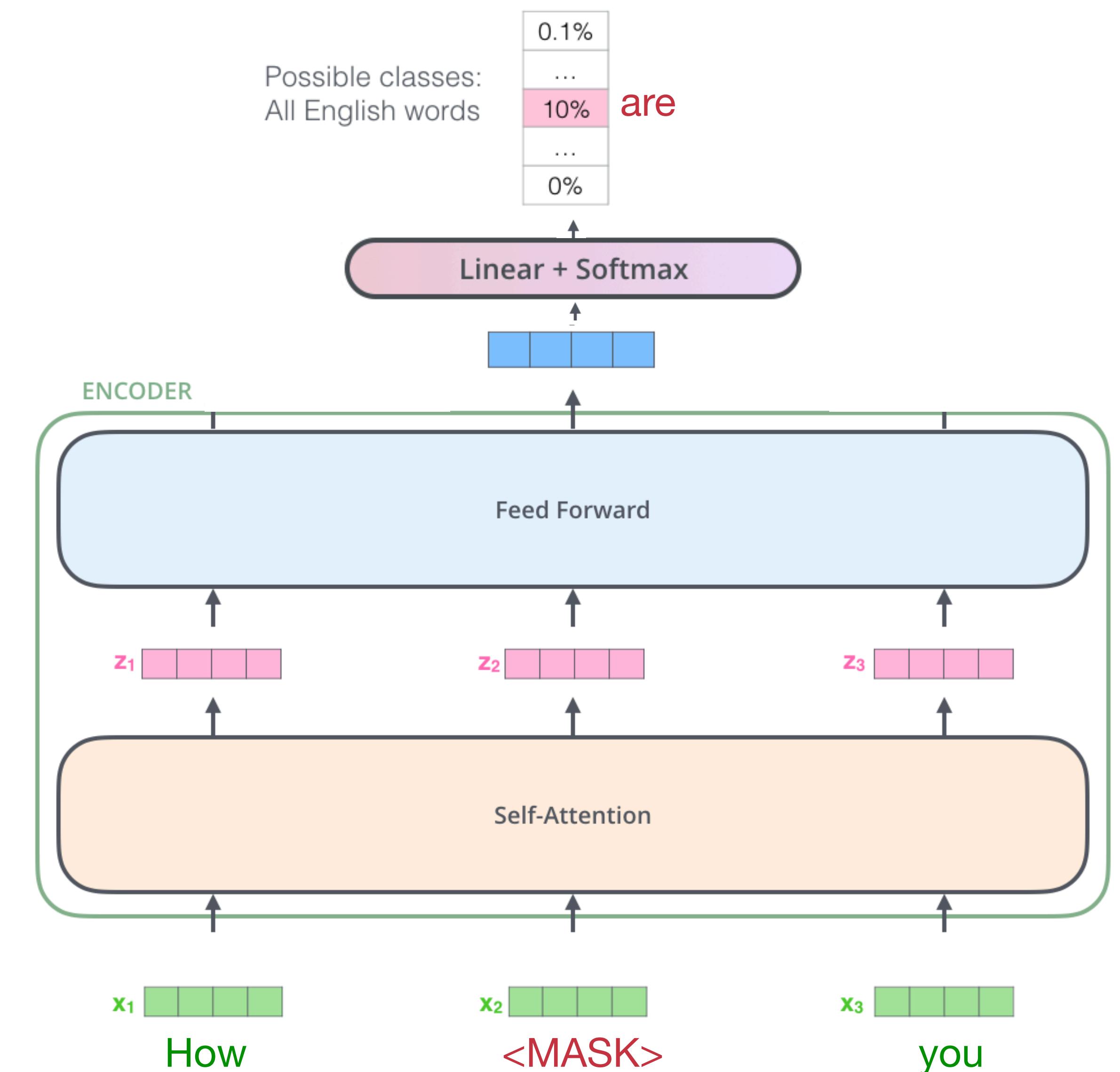
Masked Language Model

Случайно выбираем 15% позиций:

- 80% заменяем на <MASK>
- 10% заменяем на случайный токен
- 10% оставляем

Задача: предсказать исходный токен

Обученные эмбеддинги учитывают контекст слева и справа



BERT

Next Sentence Prediction

Для некоторых задач нужно понимать взаимоотношения между двумя предложениями:

- Similarity
- Entailment
- Question Answering
- ...

BERT

Next Sentence Prediction

Вход: 2 предложения (A и B)

50% - В следует за А в тексте

50% - В выбрано случайно

Формат входа:



BERT

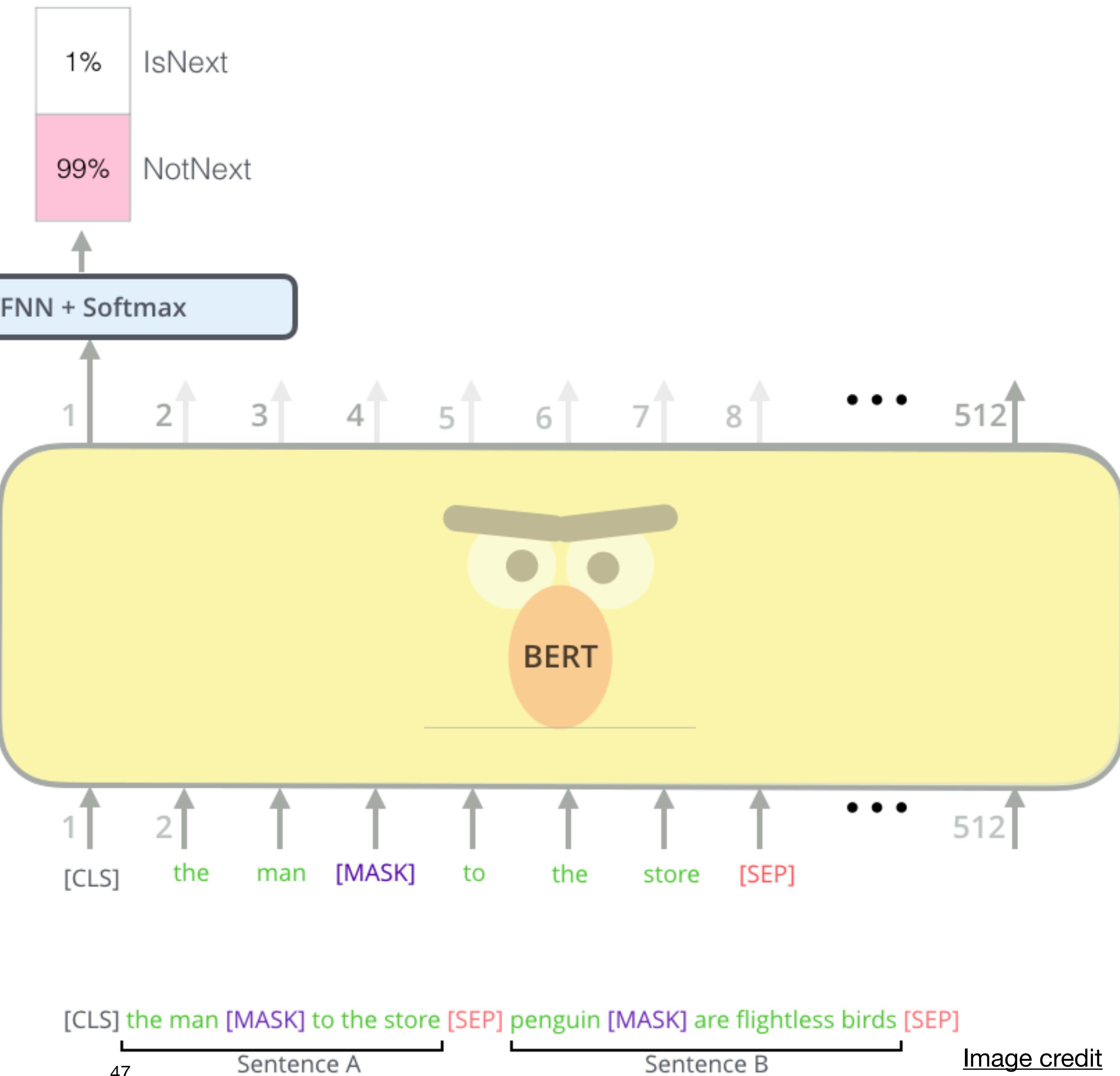
Next Sentence Prediction

Вход: 2 предложения (A и B)

50% - В следует за A в тексте

50% - В выбрано случайно

Задача: предсказать, следует ли
B за A



BERT

Next Sentence Prediction

Вход: 2 предложения (A и B)

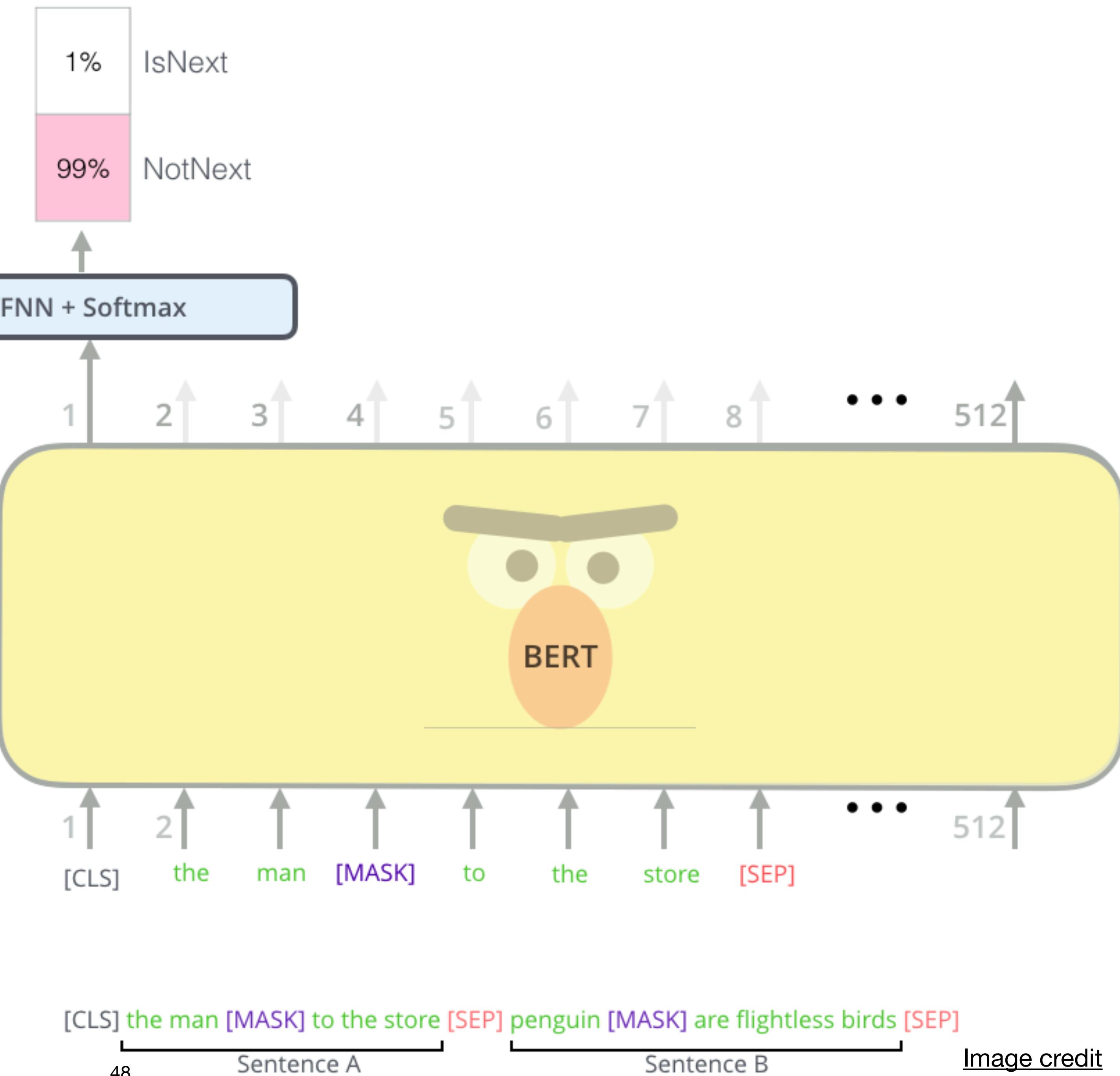
50% - В следует за A в тексте

50% - В выбрано случайно

Задача: предсказать, следует ли
B за A

<CLS> - выучивает
агрегированную информацию

<SEP> - разделитель



BERT

Как использовать?

- Linear+Softmax поверх <CLS> - для задач классификации предложения (или двух)
- Linear+Softmax поверх всех выходов - для задач классификации токенов
- Выходы BERT - как вход в другие модели (task-specific)
- ...

Retrieval-Augmented Generation (RAG)

Что такое RAG?

- RAG — это архитектура, объединяющая генеративные модели и поиск информации.
- Используется для улучшения ответов LLM за счёт доступа к внешним источникам знаний.
- Пример: ChatGPT с подключением к базе данных документов.

Зачем нужен RAG?

- Модели могут «забывать» факты — RAG помогает избежать галлюцинаций.
- Актуализация знаний без дообучения модели.
- Используется в чат-ботах, поиске по документации, поддержке клиентов и др.

Компоненты RAG

- 1. Ретривер (Retriever) — ищет релевантную информацию в базе.
- 2. Генератор (Generator) — LLM, которая формирует ответ, используя найденный контекст.
- 3. Хранилище знаний — документы, базы данных, векторное хранилище (например, FAISS).

Как работает RAG?

- 1. Пользователь задаёт вопрос.
- 2. Ретривер находит релевантные документы.
- 3. Генератор формирует ответ, используя контекст из найденных документов.

Преимущества и вызовы

- ✓ Актуальность информации
- ✓ Гибкость и масштабируемость
- ! Необходима качественная база знаний
- ! Возможные ошибки на этапе извлечения

Заключение

- RAG — эффективный способ интеграции знаний в генеративные модели.
- Позволяет моделям быть «в курсе» без переобучения.
- Важно: качество данных и настройка ретривера критичны для успеха.

Как использовать?

Большой список доступных моделей и удобный интерфейс - библиотека
HuggingFace Transformers 