



Факультет компьютерных  
наук

Аналитика больших  
данных

Москва  
2025

# Большие языковые модели: введение



1. Большие языковые модели и их архитектуры
2. Языковое моделирование и генерация текста с помощью LLM
3. Стратегии дообучения LLM под разные задачи
4. Работа с флагманскими моделями через API
5. Задачи RAG и Агенты
6. Ограничения LLM: галлюцинации, контекстное окно и другое
7. Инференс и развертывание: качество vs скорость
8. Бенчмарки и валидация





- Занятие: лекция + семинар
- Время: четверг в 18:10
- Репозиторий курса

HW	Задание	Выдача	Мягкий дедлайн	Жёсткий дедлайн
Д31	Адаптация LLM под задачу: данные → обучение → оценка + анализ ошибок	29.01.2026	13.02.2026	18.02.2026
Д32	На выбор: (A) соревнование или (B) проект "мини-ChatGPT-прототип"	19.02.2026	19.03.2026	22.03.2026

Итоговая оценка:  $0.4 * \text{Д31} + 0.6 * \text{Д32}$

Политика дедлайнов: -1 балла в день, максимум -7 за одно ДЗ.



## План занятия:

4

1. LLM верхнеуровнево
2. Ландшафт моделей: closed-source vs open-source
3. Лидерборды и бенчмарки
4. Данные для LLM
5. Prompting как интерфейс: zero-shot / one-shot / few-shot
6. Токены и токенизация
7. Мультимодальность



## Инструменты

- TF-IDF, Bag of Words, Word2Vec
- LSTM (рекуррентные сети)
- Трансформеры

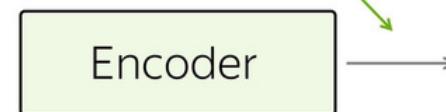


encoder      decoder      both  
BERT            GPT            T5

- язык
- задача



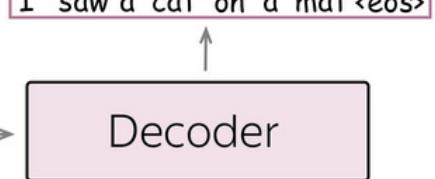
Encoder builds a representation of the source and gives it to the decoder



"I" "saw" "cat" "on" "mat"

Source sentence

Target sentence  
I saw a cat on a mat <eos>



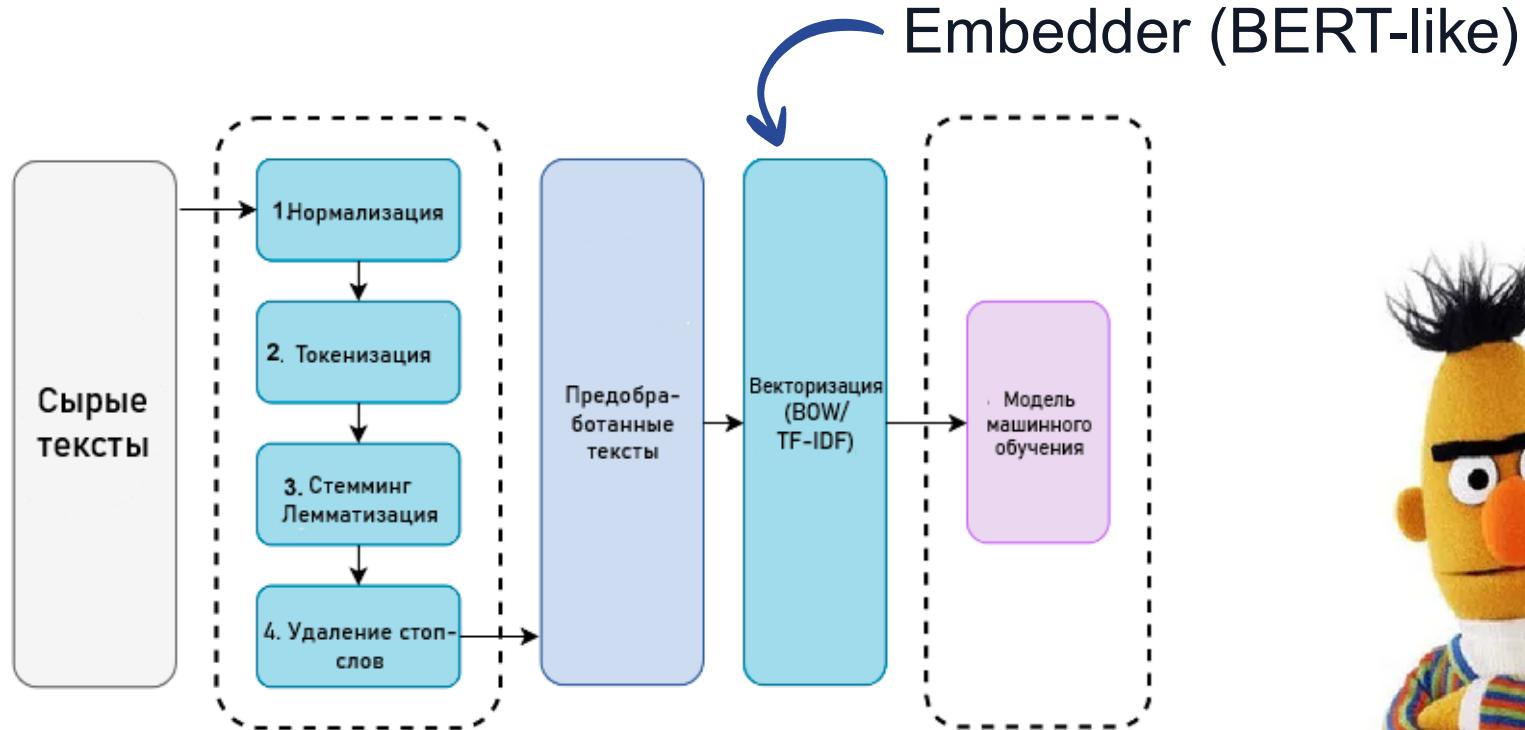
Decoder uses this source representation to generate the target sentence

Источник





# NLP-пайплайн: от текста к модели



Категория (порядок величины)	Примерный объём параметров <sup>1</sup>	Типичные возможности	Основные сценарии
Микро	≤ 1 В (сотни М)	Ответы на FAQ, простая классификация, keyword-RAG	Смартфоны, микроконтроллеры, IoT-устройства
Малые	1 – 10 В	Диалог на бытовые темы, тонкая дообучаемость, базовый код-ассистент	Edge-серверы, офлайн-мобильные ассистенты, встраивание в приложения
Средние	10 – 30 В	Полноценный чат-бот, краткое резюмирование, генерация кода средней сложности	Корп. микросервисы, ноутбук с одной GPU, кастом-RAG
Крупные	30 – 100 В	Длинный контекст (32 К+), аналитические ответы, многоговорящий, устойчивые цепочки рассуждений	SaaS-боты с высоким качеством, инструменты разработчика, internal-search
Очень крупные	100 – 300 В	SOTA-точность на NLU/NLG бенчмарках, сложные запросы «пишем код + тесты», базовая мультимодальность	Чат-ассистенты в продуктах, сводные отчёты, корпоративная аналитика
Frontier / «Триллионники»	> 300 В или MoE-экв.	Продвинутая мультимодальность (текст-картина-аудио-видео), агентное планирование, научные рассуждения	R&D-лаборатории, высокоуровневые агенты, генерация длинных документов

<sup>1</sup> Dense-параметры; Mixture-of-Experts указывается как «активные» параметры (т.е. реально вычисляемые на проход).



## Closed-Source модели

- OpenAI API (через VPN) - [[openai.com/api](https://openai.com/api)]
- YandexGPT Lite/Pro (поддерживает дообучение) - [[console.yandex.cloud](https://console.yandex.cloud)]
- GigaChat API (без дообучения) - [[developers.sber.ru](https://developers.sber.ru)]

## Open-Source модели

- [[ruGPT-3.5](#)]
- [[GigaChat-20B-A3B-instruct](#)]
- [[Meta-Llama-3-8B](#)] (требует HF token)
- [[Mistral-7B-Instruct](#)] (требует HF token)
- [[Qwen3-8B](#)]
  - DeepSeek-LLM-7B-Instruct
  - Gemma-3-4b-it

В Мире

LLM		
P:	Gemini	ChatGPT
		Grok
		Claude
OS:	deepseek	
	Qwen	Meta AI LLaMA
	MISTRAL AI	Yi
В РФ		
P:	GIGA CHAT	Y
		M T AI
OS:	t-tech/T-pro-it-1.0	
	IlyaGusev/saiga_nemo_12b	



## Actual in 2025:

- **Qwen 3** — strong multilingual + reasoning
- **DeepSeek V3 / R1** — efficient reasoning-optimized release
- **Kimi K2, GLM-4** — Chinese-English bilingual models
- **Mistral / Mixtral** — 7B dense and 8x7B MoE models for speed/quality trade-off
- **LLaMA-3,4**
- **gpt-oss**

Quantization (4–8 bit) makes these fit on consumer GPUs.

Most integrated in Transformers and vLLM runtimes.

## Actual in 2024:

- **LLaMA-3** 8B – 13B – 70B – [405B]
- Mixtral, Qwen, Qi, Mistral(for smaller sizes)

## Actual in 2022-2023:

- **LLaMA-2** – good first choice for English tasks (and some other languages) 7 – 13- 70 B
- **Falcon-180B** – somewhat better, but a lot larger (7B, 40B and 180B parameters)
- **BLOOM-176B** – if the first two models don't speak your language (560M, 7B, 176B parameters)



# Как выбрать SOTA - го на арену

10

Chatbot Arena [[chat.lmsys.org](https://chat.lmsys.org)] - они же на hf

Hugging Face - [open-llm-leaderboard](#)

GLUE / SuperGLUE

Russian SuperGLUE

MTEB (Massive Text Embedding Benchmark)



# LLM benchmarking @ Chatbot Arena (2025)

11

<https://huggingface.co/spaces/lmarena-ai>

Data as of 2025-10-13

Rank (UB)	Model	Score
1	gemini-2.5-pro	1452
1	claude-sonnet-4-5-thinking-32k	1448
1	claude-opus-4-1-thinking-16k	1448
2	chatgpt-4o-latest	1441
2	gpt-4.5-preview	1441
2	gpt-5-high	1440
2	o3	1440
3	qwen3-max-preview	1434
9	glm-4.6	1421
9	grok-4-fast	1420
9	deepseek-v3.2-exp-thinking	1418
11	kimi-k2-0905-preview	1416
11	qwen3-235b-a22b-instruct	1418



GitHub для машинного обучения

### Ключевые продукты:

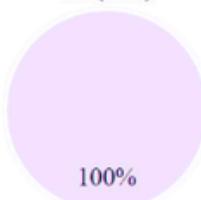
- 🤗 Transformers (первая версия — 2019) [huggingface.co](https://huggingface.co)
- Hub (Models / Datasets / Spaces)
- Библиотеки Datasets, Diffusers, PEFT, Accelerate, Evaluate, Optimum
- Gradio / Spaces — no-code витрины моделей
  
- 1 786 000+ публичных моделей [huggingface.co](https://huggingface.co)
- 421 000+ датасетов [huggingface.co](https://huggingface.co)
- ≈ 200 000 демо-приложений (Spaces) [originality.ai](https://originality.ai)
- 50 000+ организаций-контрибьюторов



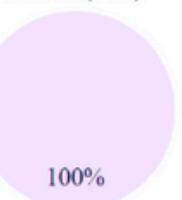
# Data for LLMs: domains and sources

13 |

T5 (11B)



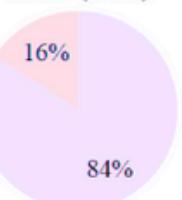
Falcon (40B)



LLaMA (65B)



GPT-3 (175B)



MT-NLG (530B)



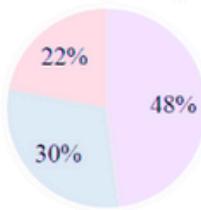
Gopher (280B)



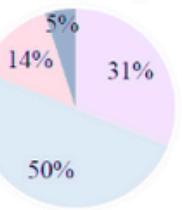
Chinchilla (70B)



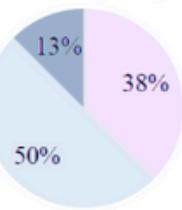
GLaM (1200B)



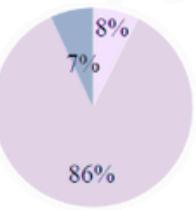
PaLM (540B)



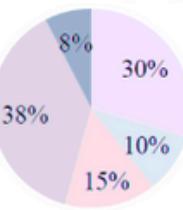
LaMDA (137B)



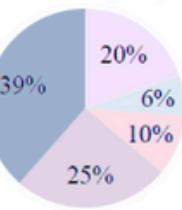
Galactica (120B)



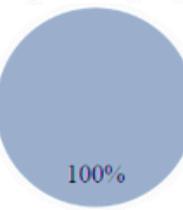
GPT-NeoX (20B)



CodeGen (16B)



AlphaCode (41B)



Webpages

C4 (800G, 2019), OpenWebText (38G, 2023), Wikipedia (21G, 2023)

Conversation Data

the Pile - StackExchange (41G, 2020)

Books & News

BookCorpus (5G, 2015), Gutenberg (-, 2021), CC-Stories-R (31G, 2019), CC-NEWES (78G, 2019), REALNEWS (120G, 2019)

Scientific Data

the Pile - ArXiv (72G, 2020), the Pile - PubMed Abstracts (25G, 2020)

Code

BigQuery (-, 2023), the Pile - GitHub (61G, 2020)



Идея: использовать весь “чистый Интернет”.

Реальность: Интернет грязный и плохо отражает то, что нам нужно.

Практика:

- Скачать весь Интернет. Например, Common Crawl: 250 млрд страниц, >1 ПБ данных (>1e6 ГБ).
- Извлечение текста из HTML (сложности: формулы/математика, шум вроде шаблонных блоков).
- Фильтрация нежелательного контента (например, NSFW, вредный контент, персональные данные).
- Дедупликация (по URL/документу/строкам). Например, в форумах шапки/меню часто полностью одинаковые.
- Эвристическая фильтрация: удалять низкокачественные документы (например, по числу слов, средней длине слов, выбросам по токенам).
- Фильтрация с помощью моделей: предсказывать, может ли страница быть достойным источником (например, как у Википедии).
- Смешивание данных: классифицировать источники по типам (код/книги/развлекательные тексты и т.п.).
- Перевзвешивание доменов с учётом законов масштабирования, чтобы получить более высокое качество на прикладных задачах.

Источник: CS229



# The Prompting Paradigm

	zero-shot	one-shot	two-shot
Input (prompt)	Review: I love this movie! Sentiment:	Review: This movie sucks. Sentiment: negative	Review: This movie sucks. Sentiment: negative  Review: This was cool! Sentiment: positive
Model output	positive	positive	positive

# Токенизация



## Токен - единица текста, с которой работает модель (не обязательно слово).

17

- Почему это критично:
- стоимость запроса и скорость - в токенах
- длина контекста - в токенах
- разные токенизаторы по-разному режут текст
- редкие/неудачные токены могут ломать качество

tokenization

token

ization

Текст → токены → числа (ID) → эмбеддинги

### Практический вывод

Все лимиты и стоимость удобно обсуждать в токенах: вход/выход, максимум контекста, а также влияние настроек генерации на длину и стабильность ответа.



# Токенизация

18

Токенизация на уровне символов

```
["I", " ", "❤", " ", "N", "L", "P", "!", " ", "|", "t", "", "s", " ", "s", "o", " ", "m", "u", "c", "h", " ", "f", "u", "n", "!", " ", "😎"]
```

Токенизация на уровне слов

```
["I", "❤", "NLP", "!", "It's", "so", "much", "fun", "!", "😎"]
```

Токенизация по пробелам

```
["I", "❤", "NLP!", "It's", "so", "much", "fun!", "😎"]
```

Токенизация с учётом  
знаков препинания

```
["I", "❤", "NLP", "!", "It", "", "s", "so", "much", "fun", "!", "😎"]
```

Токенизация через  
кодирование пар байтов  
Byte-Pair Encoding (BPE)

```
["I", "❤", "N", "LP", "!", "It", "", "s", "so", "much", "fun", "!", "😎"]
```

"much", "fun", "!", "😎"]

"I ❤ NLP! It's so much fun! 😎"



# Byte-Pair Encoding

“newer never”

19

Алгоритм сжатия текста, который используется в токенизации для уменьшения размера словаря и обработки редких слов.

1. Разбиение на символы
2. Подсчёт частоты пар
3. Слияние пар
4. Повторение

n e : 2 раза

e w : 1 раз

e r : 2 раза

n v : 1 раз

v e : 1 раз

n e w e r n e v e r

n e w e r n e v e r

["ne", "w", "e", "r", "ne", "v", "e", "r"].



“apple” -> ["app", "le"]

“unhappiness” -> ["un", "happiness"]

- Сжатия текста

Слияние частых символов позволяет эффективно уменьшить размер словаря, особенно для редких слов.

- Универсальности

Позволяет работать с редкими или неизвестными словами, разделяя их на более мелкие, часто встречающиеся компоненты.

- Улучшения обработки OOV (out-of-vocabulary)

Слова, которых нет в словаре, можно разобрать на более мелкие токены, что позволяет эффективно справляться с новыми или редкими словами.



# Токенизация

21

Нормализация  
юникод/пробелы  
нижний регистр (опц.)

Предтокенизация  
грубое разбиение  
на «слова»/символы

Подслова  
BPE / WordPiece  
/ Unigram

Выход: токены и их числовые ID (+ маски, паддинг, спец-токены)

## Мини-пример:

Текст: "I love tokenization!"

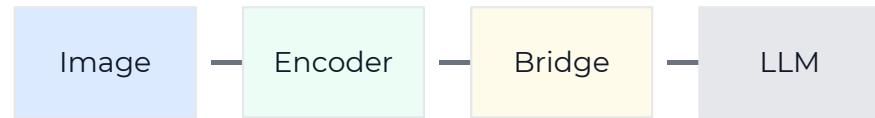
Токены: ["I", "love", "tokenization", "!"]

ID: [ 40, 1842, 19233, 1634, 0 ]

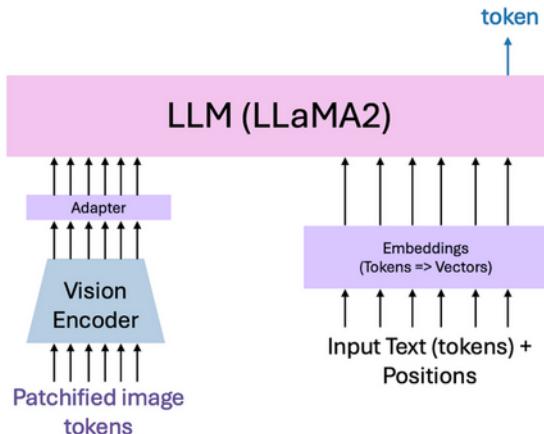
визуальный энкодер → мост (проекция/адаптер) → LLM.

Три типовых паттерна:

- замороженный энкодер + замороженная LLM + адаптер
- кросс-внимание к визуальным признакам
- мультимодальное дообучение по инструкциям (чат)



Идея: превращаем изображение в визуальные токены и подаём в LLM как контекст.



Пример: LLaVA  
Визуальные признаки → проекция → визуальные токены → LLM

