



Домашний трансформер

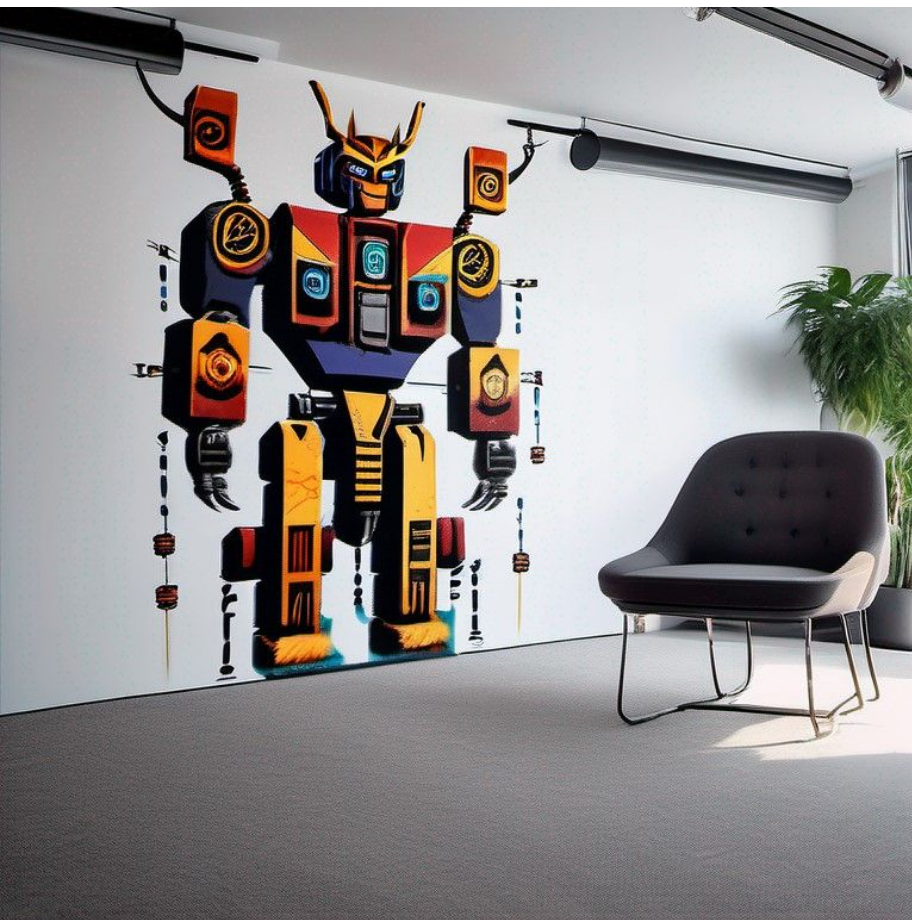
Как завести себе дома

маленький аналог
chat-gpt

Что сегодня будет

- **Свежие** открытые модели
- Оболочки
- Как поднять
- Железо vs скорость
- Квантование
- Как настроить





Чего сегодня не будет

- Платных сервисов
- Формул
- Мастер-класса

Свежие open-source модели

Если нужен english:

- Mistral, LLama-2
- Falcon

Русский тоже бывает

- Saiga
- ruGPT

https://huggingface.co/IlyaGusev/rpr_7b





LLama-2, Июль 2023, Meta

Варианты модели:

- 7B, 13B, 70B

Хитрая лицензия, Качество работы топ

Falcon, Сентябрь 2023, TII, UAE

Варианты модели:

- 180B, 40B, 7.5B, 1.3B

На текущий момент самая
большая открытая модель

По качеству работы между
GPT-3.5 и GPT-4



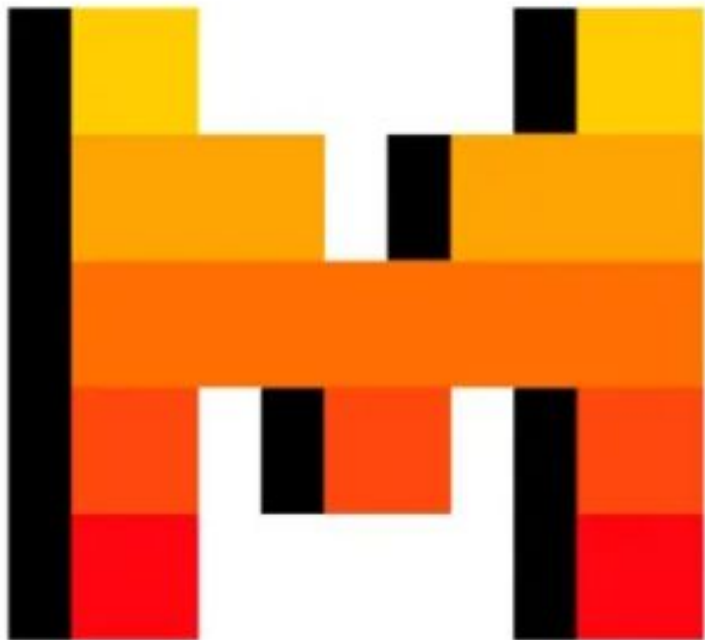


Saiga-2,

Доученная на русских диалогах
LLama-2

Проект сообщества

https://huggingface.co/IlyaGusev/saiga2_7b_lora



Mistral, Сентябрь 2023

Новая модель от стартапа из Франции

Доступна только 7B версия

Лицензия Apache 2.0

По бенчмаркам бьет ламу 2,
соревнование продолжается

Как измерять модели

Число параметров,

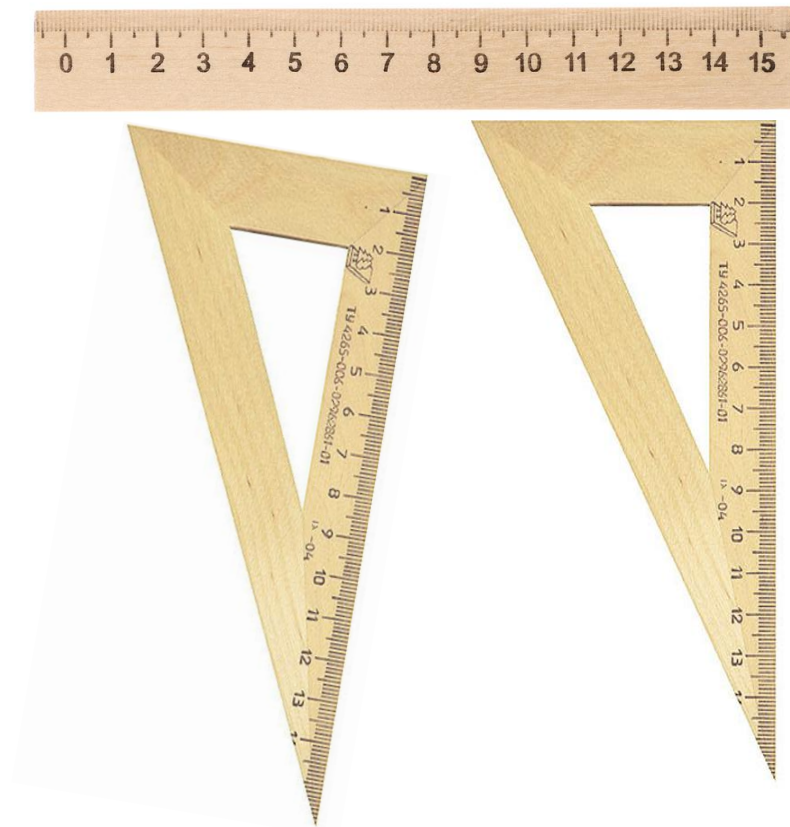
Чем “больше” параметров, тем в *целом* модель лучше.

Жирные модели дорогие для прода.

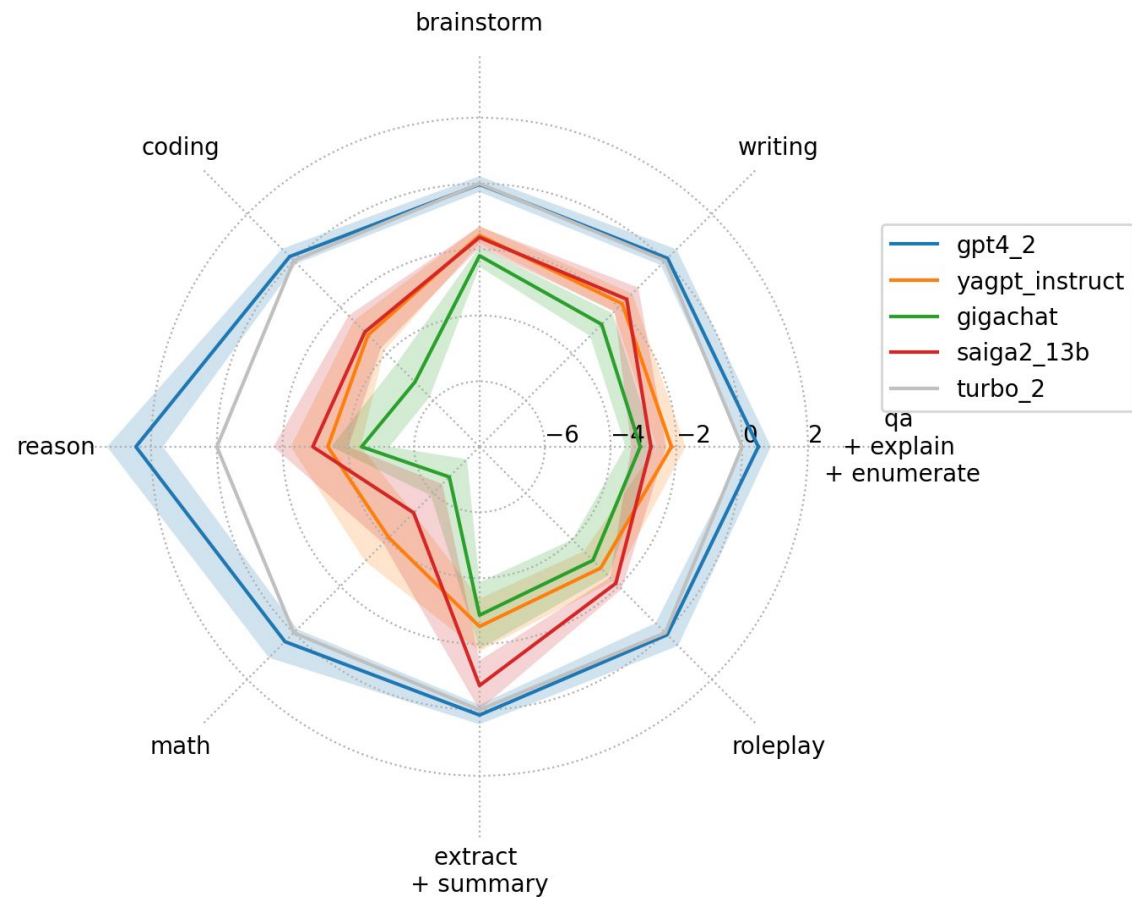
Downstream метрики,

Лучше всего модель мерять на тех задачах под которую ее готовили,

Например сдать своеобразный ЕГЭ



Downstream метрики определенных моделей



Сколько, сколько гига?

один Параметр, занимает

4 байта (32 бита) это *float* из C++, C#

Для нейросетей, такая точность считается *избыточной*.

7В в названии модели в таком режиме будет занимать $7 * 4 = 28$ Гб ОЗУ

но не все так плохо,

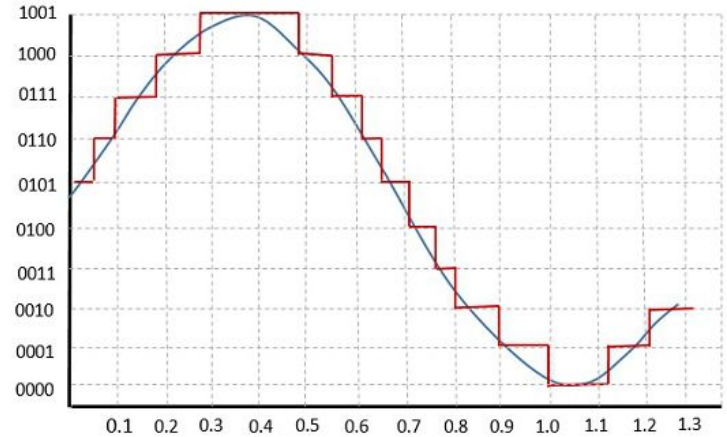


Сжимаемся

2 байта (16 бит)/параметр – в таком формате нет потери качества

Есть только на GPU, и самых свежих процах

Все что меньше требует *квантования*, и уже есть потери



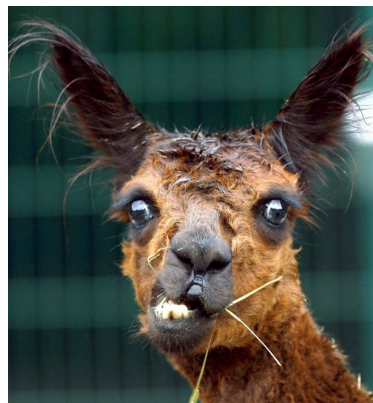
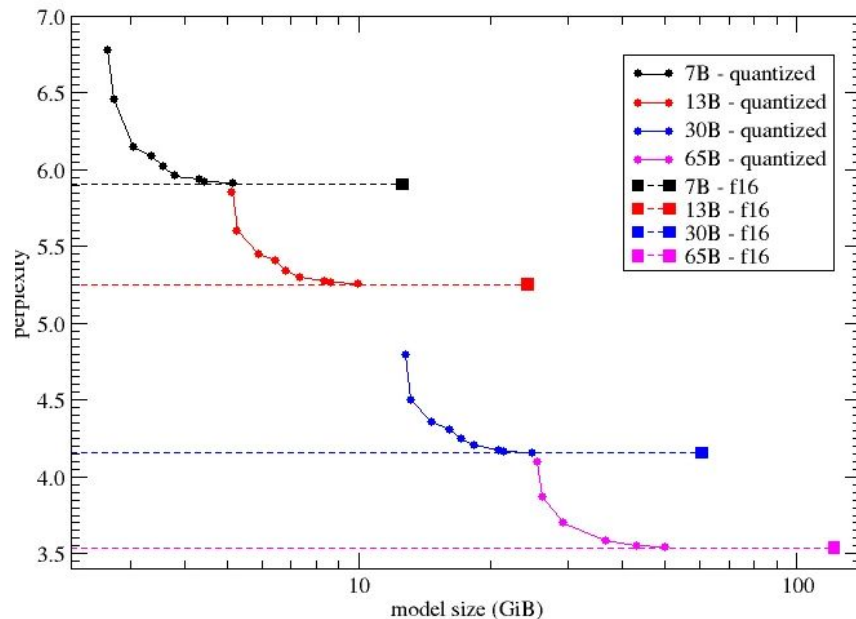
Квантование

Можно сжать даже до 1 бита, но с существенной потерей качества

2 бита/параметр дают точность, сопоставимую с моделью с меньшим числом параметров.

То есть есть профит ужимать до 4 бит, меньше смысла *мало*.

Возможно, все станет лучше, с прокачкой методов квантования.



Длина контекста

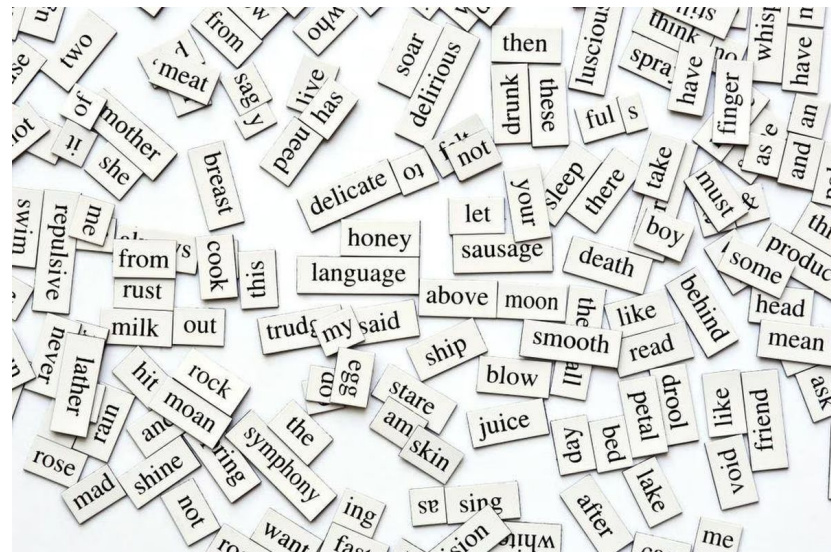
	Модель	Длина контекста	Страниц текста на english
Насколько длинный текст (число токенов) можно подать на вход модели.	GPT-3.5	4,096	6
	GPT-4	8,192	12
	GPT 4-32k	32,768	49
Для чата это длина всего диалога	Llama-1	2,048	3
	Llama-2	4,096	6

Токены

Для того чтобы модель съела текст, его нужно нарезать.

Можно резать, *по-словам, по-буквам, но так уже никто не делает.*

Все современные модели используют, токенизацию(нарезку) на под слова.



Подслова (subword tokenization)

Нарезка происходит исходя из частотности и фиксированного размера словаря.

Если целое слово в словарь не влезит, то в словарь попадут части слов

Словарь обычно лежит в папке рядом с весами

Tokens	Characters
78	389

The GPT family of models process text using tokens, which are common sequences of characters found in text. The models understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text would be tokenized by the API, and the total count of tokens in that piece of text

TEXT TOKEN IDS

Токенизация русского языка

Если модель учили не на русском, то русский язык может обрабатываться очень не эффективно.



Привет токенизация русского у GPT-3

У OpenAI есть визуализатор токенизации
<https://platform.openai.com/tokenizer>

Как можно поднять

Популярные запуски

- Huggingface (CPU, **GPU**)
- lama.cpp (**CPU**, GPU)



Локальный ui

Training Session

Write a Python script that creates a temporary file, writes a random 16-character password to this file, and then prints its path. Make your reply short as I need it to fit into a screenshot.

Sure! Here's a short Python script that does what you described:

```
import os
import random

# create a temporary file
with tempfile.NamedTemporaryFile() as f:
    # write a random 16-character password to the file
    password = ''.join(random.choice('23456789abcdefghijklmnopqrstuvwxy') for _ in range(16))
    f.write(password.encode())

# print the path of the temporary file
print(f.name)
```

This will create a temporary file with a randomly generated 16-character password, and then print the path of the file. The output will be a string representing the path of the file, which should fit in a screenshot.

You forgot to import `tempfile`.

Oops, you're right! Here's the corrected code with the `tempfile` module imported:

```
import os
import random
from tempfile import NamedTemporaryFile

# create a temporary file
```

Input

|

Stop Generate Continue

Impersonate Regenerate Remove last

<https://github.com/oobabooga/text-generation-webui>

Где качать модели

Почти самый главный “пират”

Выкладывает готовые модели с
разной степенью квантования



<https://huggingface.co/TheBloke>



Требования к железу

Желательно чтобы параметры
влезали в оперативку.

Если хочется быстро, то в оперативку
видяхи

Если, не влезает. то придется много
читать с диска.

Насколько быстро все это работает

Перформанс меряется в токенах в секунду, или секунд на токен

Большие модели отвечают медленно, требуют больше железа,

Экономика больших моделей для прода, все еще очень большая проблема



Волшебные рычаги

- Температура
- Top-k/Top-p
- Beam-search
- Штрафы за повторы n-gram
- Штраф за длину ответа
- Управляемая генерация

Дерево ответов

Языковые модели выдают
“вероятные продолжения”.

Мы выбираем один жадно или
используем Top-p, Top-k
сэмплирование

Так-же можно юзать Beam-search
и Штрафы



Температура

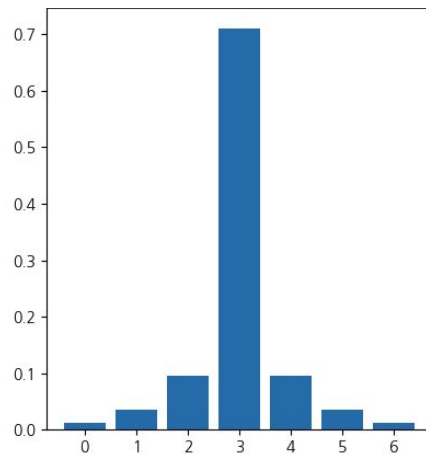
Задаёт вариативность ответа, по факту мы кидаем кубики

Большая температура подходит когда от модели требуется больше фантазии

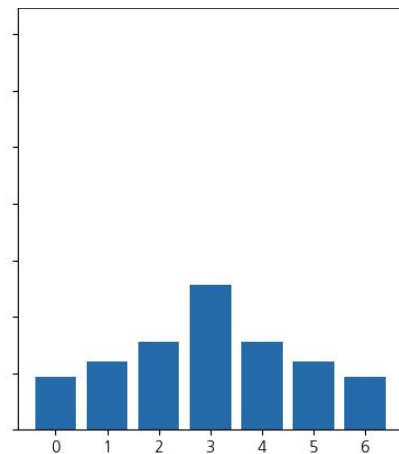
Нулевую температуру ставят когда требуется правильный ответ.

Это “жадное” поведение

Варианты ответов



T = 0



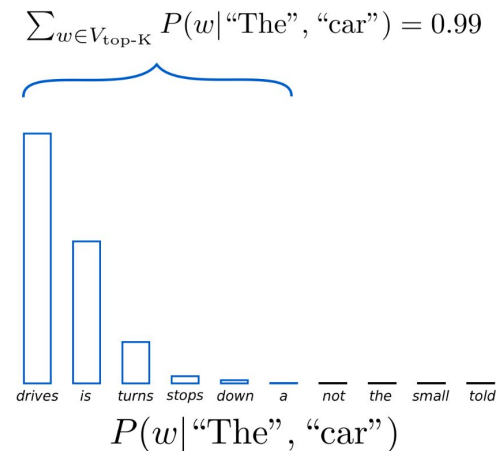
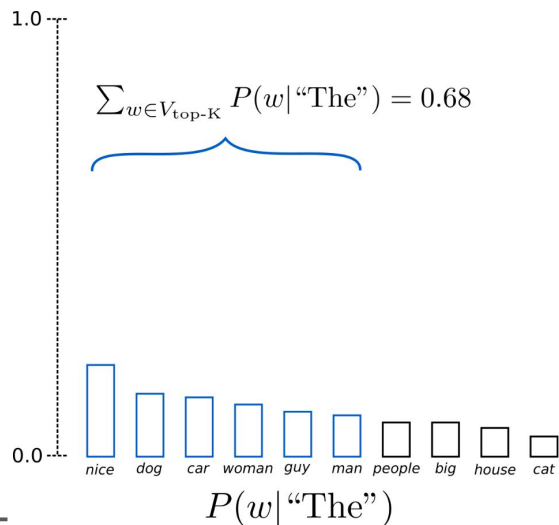
T = 10

Top-k/Top-p

k = 4 - Самых лучших вариантов

p = 90% - минимальное множество вариантов, суммарная вероятность которых **больше** p

Такой способ еще называют **nucleus-sampling**

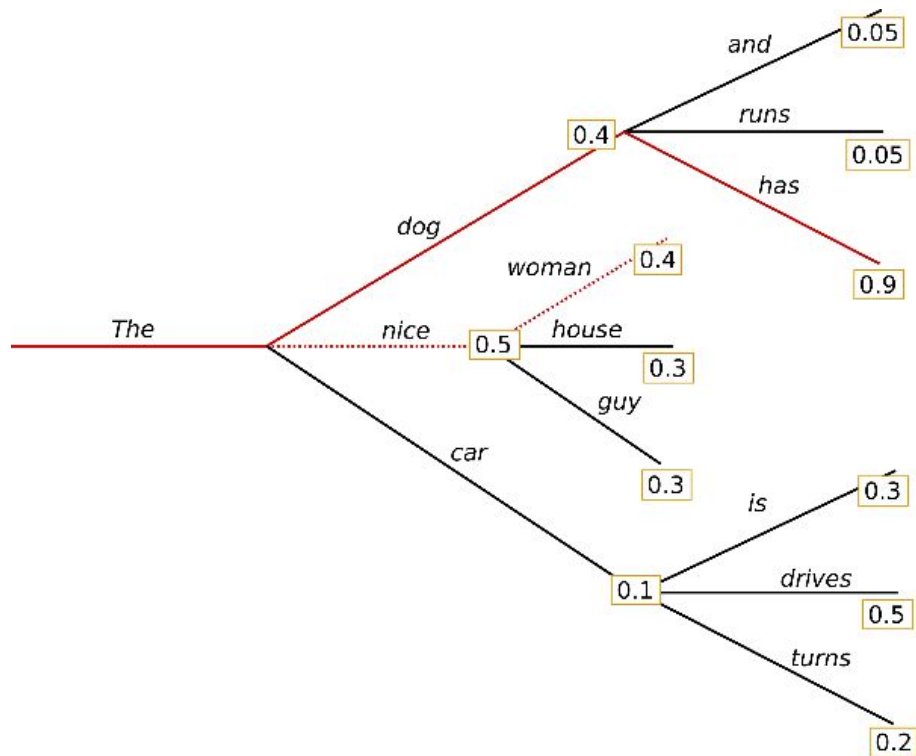


Beam search

Идем сразу по n самым вероятным ветвям, в конце выбираем лучшую

Позволяет получить более качественные и вариативные ответы

$n > 5$ не имеют смысла, плюс, все это работает медленнее



Штрафы

Иногда модели **зацикливаются**,

Решение делаем beam-search с **штрафом** на повторы, токенов, пар токнов, n-токенов (1-gram, 2-gram, n-gram)

Также можно штрафовать за длинные и короткие ответы.

Управляемая генерация

Когда: Если от модели ожидается ответ в каком то формате - код, или json

Мы просто игнорируем варианты,
которые не вписываются в формат ответа

That's all folks!

Кто хотел может убежать

- Вопросы?

Дальше будет только флуд

