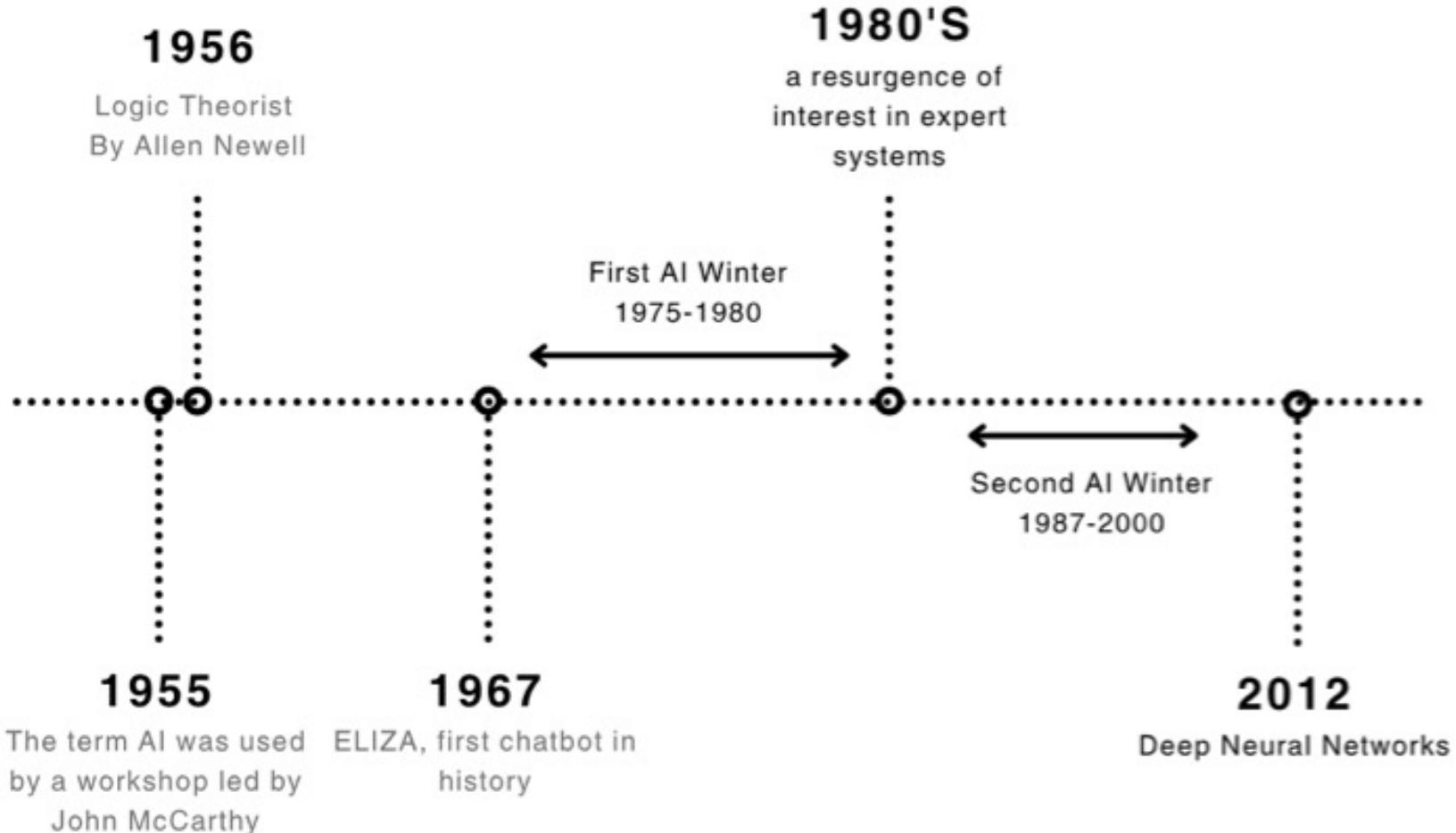


Языковые модели в ИИ

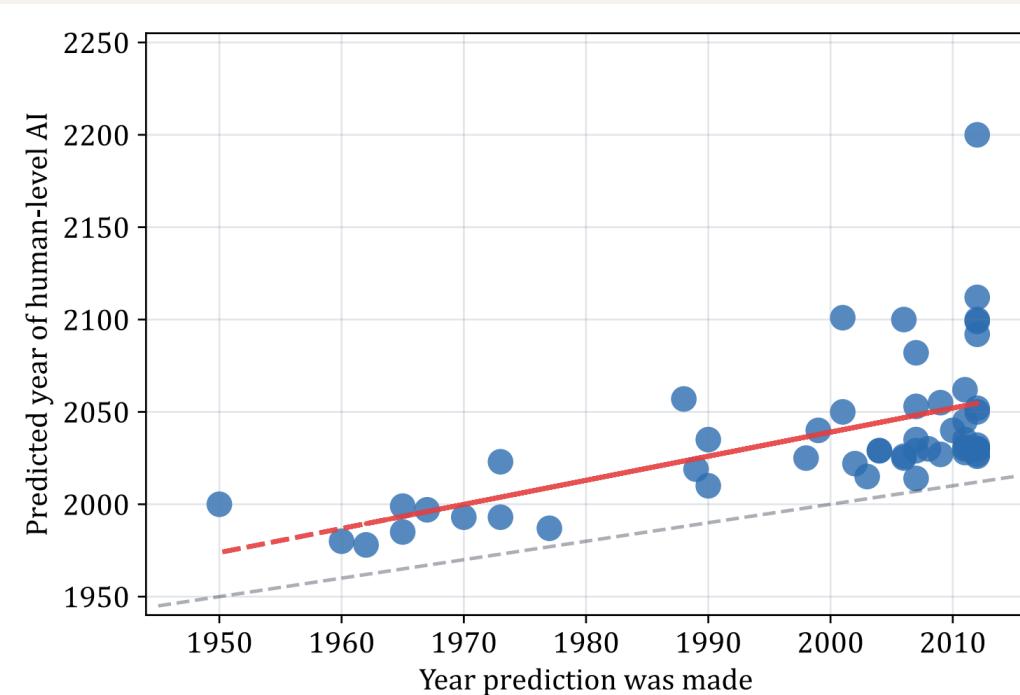
Владимир Судаков
2025

Развитие ИИ



Прогноз «сильного» ИИ

“decade after decade, AGI remained roughly 25 years on the horizon”



Мешок слов

Для языковой модели нужно превращать текст с числа, чтобы мы могли их обработать.

Bag-of-Words

[**"a", "and", "are", "discovery", "enjoy",
"everyone", "folk", "for", "fun",
"great", "important", "interesting", "is",
"learning", "listen", "math",
"movie", "movies", "music", "research",
"rock", "science", "to", "today",
"very", "watching"]**

1. Movies are fun for everyone.
2. Watching movies is great fun.
3. Enjoy a great movie today.
4. Research is interesting and important.
5. Learning math is very important.
6. Science discovery is interesting.
7. Rock is great to listen to.
8. Listen to music for fun.
9. Music is fun for everyone.
10. Listen to folk music!

Document-term matrix (DTM)

Bag-of-Words: DTM

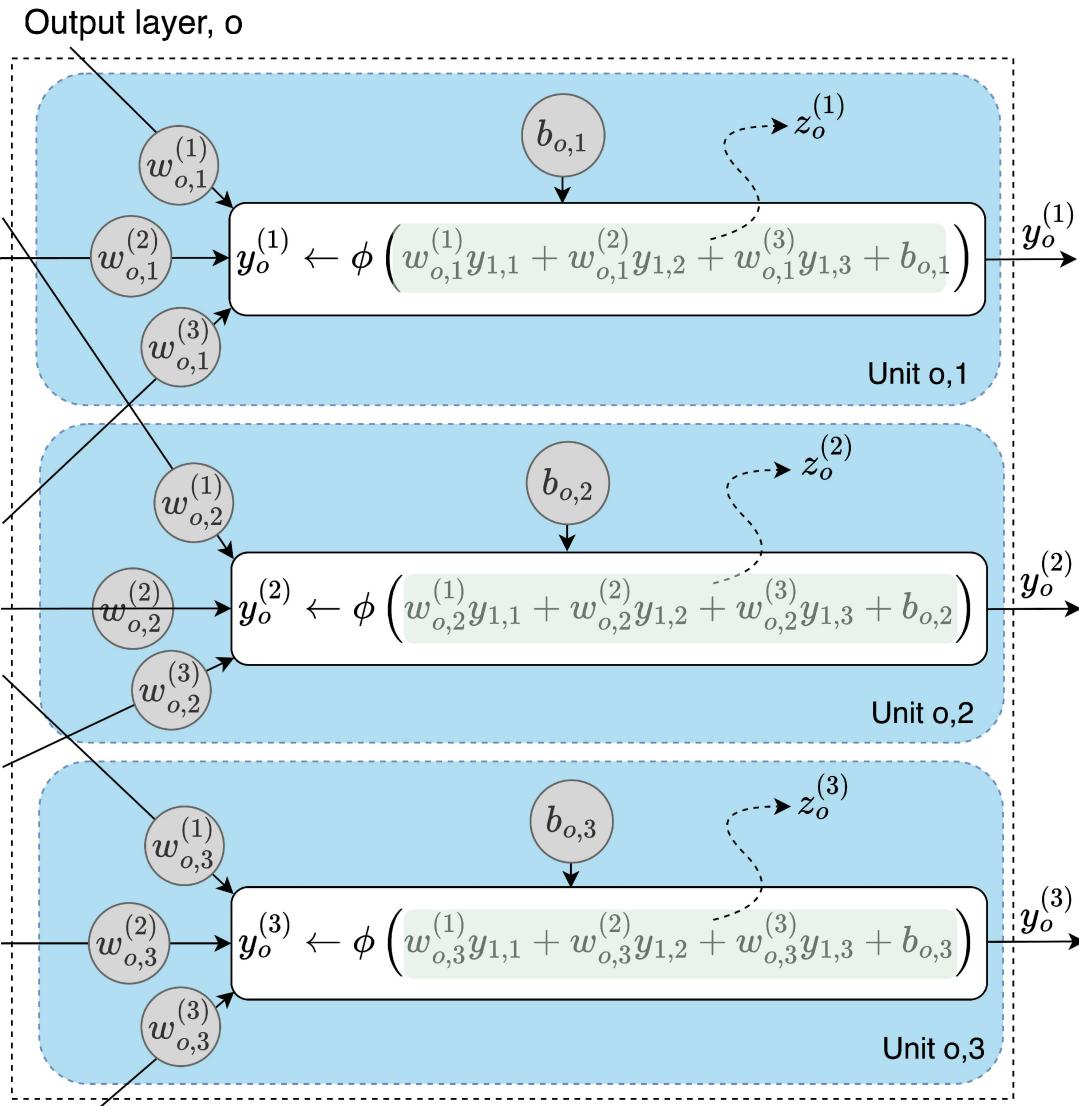
Doc	a	and	...	fun	...	listen	math	...	science	...	watching
1	0	0	...	1	...	0	0	...	0	...	0
2	0	0	...	1	...	0	0	...	0	...	1
3	1	0	...	0	...	0	0	...	0	...	0
4	0	1	...	0	...	0	0	...	0	...	0
5	0	0	...	0	...	0	1	...	0	...	0
6	0	0	...	0	...	0	0	...	1	...	0
7	0	0	...	0	...	1	0	...	0	...	0
8	0	0	...	1	...	1	0	...	0	...	0
9	0	0	...	1	...	0	0	...	0	...	0
10	0	0	...	0	...	1	0	...	0	...	0

Классификация

Doc	Text	Class ID	Class Name
1	Movies are fun for everyone.	1	Cinema
2	Watching movies is great fun.	1	Cinema
3	Enjoy a great movie today.	1	Cinema
4	Research is interesting and important.	3	Science
5	Learning math is very important.	3	Science
6	Science discovery is interesting.	3	Science
7	Rock is great to listen to.	2	Music
8	Listen to music for fun.	2	Music
9	Music is fun for everyone.	2	Music
10	Listen to folk music!	2	Music

$$\text{softmax}(\mathbf{z}, k) \stackrel{\text{def}}{=} \frac{e^{\mathbf{z}^{(k)}}}{\sum_{j=1}^D e^{\mathbf{z}^{(j)}}}$$

Архитектура выходного слоя



Функция потерь

$$\text{loss}(\tilde{\mathbf{y}}, \mathbf{y}) = - \sum_{k=1}^C y^{(k)} \log(\tilde{y}^{(k)}),$$

$$\text{loss} = -\frac{1}{N} \sum_{i=1}^N \log(\hat{y}_i^{(c_i)})$$

Ограничение мешка слов

- Игнорирует порядок слов
- Отсутствие контекстного понимания
- Разреженные и многомерные векторы
- Не может обрабатывать слова, отсутствующие в словаре

Word2Vec

Skip-gram

professor alan * research advanced

alan turing's * advanced computer

turing's research * computer science

Skipped word

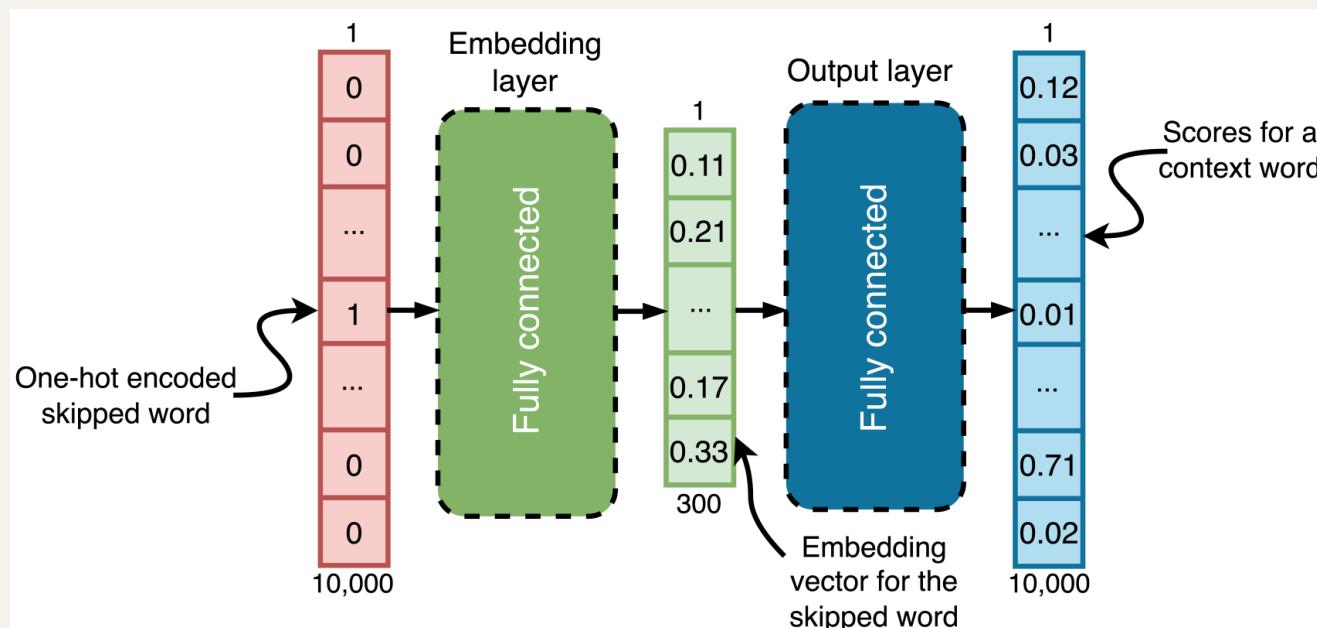
turing's

research

advanced

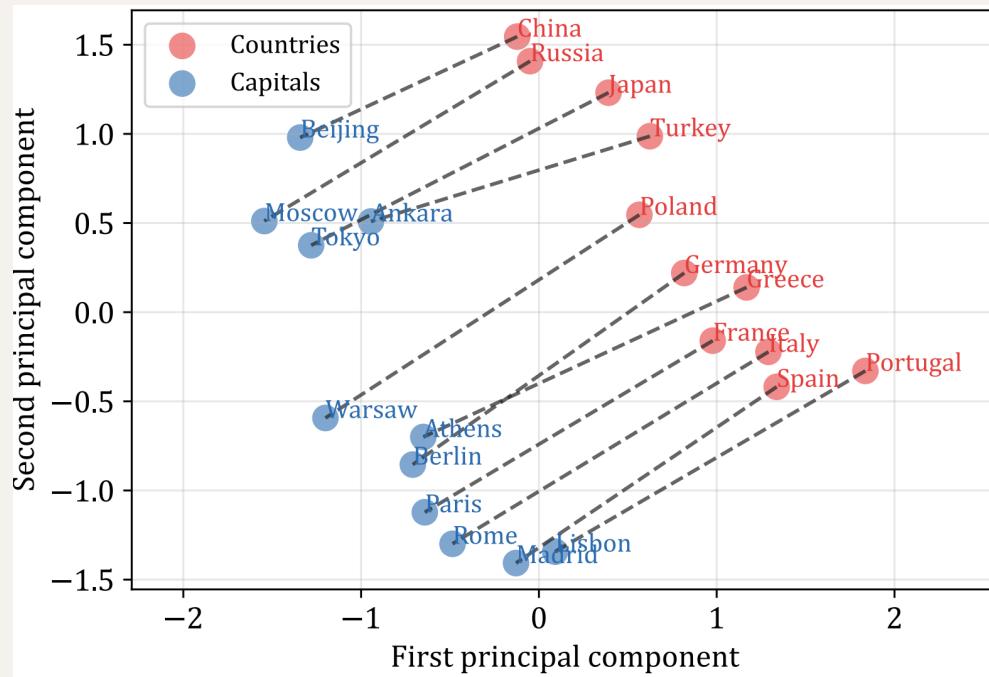
Эмбеддинги

Word Embedding



Эмбеддинги

Word Embedding: Semantic Similarity



- 200-dimensional projected by PCA
- King – man + woman = queen!

Языковая модель

A Language Model

$$P(w^1, w^2, \dots, w_n) = \\ P(w_1) \cdot P(w_2 | w_1) \cdot P(w_3 | w_1, w_2) \cdot \dots \cdot P(w_n | w_1, \dots, w_{n-1})$$

- This is a discrete probability distribution



Авторегрессионная модель

$$\Pr(t = \text{are} | s = (\text{language}, \text{models}, \text{are})) = 0.01$$

$$\Pr(t = \text{cool} | s = (\text{language}, \text{models}, \text{are})) = 0.77$$

$$\Pr(t = \text{language} | s = (\text{language}, \text{models}, \text{are})) = 0.02$$

$$\Pr(t = \text{models} | s = (\text{language}, \text{models}, \text{are})) = 0.15$$

$$\Pr(t = \text{useless} | s = (\text{language}, \text{models}, \text{are})) = 0.05$$

Этот тип модели представляет собой авторегрессивную языковую модель, также известную как причинно-следственная языковая модель. Авторегрессия предполагает предсказание элемента в последовательности, используя только его предшественников. Такие модели отлично подходят для генерации текста и включают в себя языковые модели чата на основе трансформеров (chat LMs).

В отличие от них, маскированные языковые модели, такие как BERT используют другой подход. Эти модели предсказывают намеренно замаскированные токены в последовательностях, используя как предшествующий, так и последующий контекст. Этот двунаправленный подход особенно подходит для таких задач, как классификация текста и распознавание именованных сущностей.

До того, как нейронные сети стали стандартом для языкового моделирования, традиционные методы основывались на статистических методах. Эти модели, основанные на подсчете, до сих пор используемые в автозаполнении смартфонов, оценивают вероятность последовательностей слов на основе подсчета частоты слов или n-грамм, полученных из корпуса. Чтобы лучше понять эти методы, давайте реализуем простую языковую модель, основанную на подсчете.

Триграммы

Count-Based Language Model:
A Trigram

$$P(w_i \mid w_{i-2}, w_{i-1}) = C(w_{i-2}, w_{i-1}, w_i) / C(w_{i-2}, w_{i-1})$$



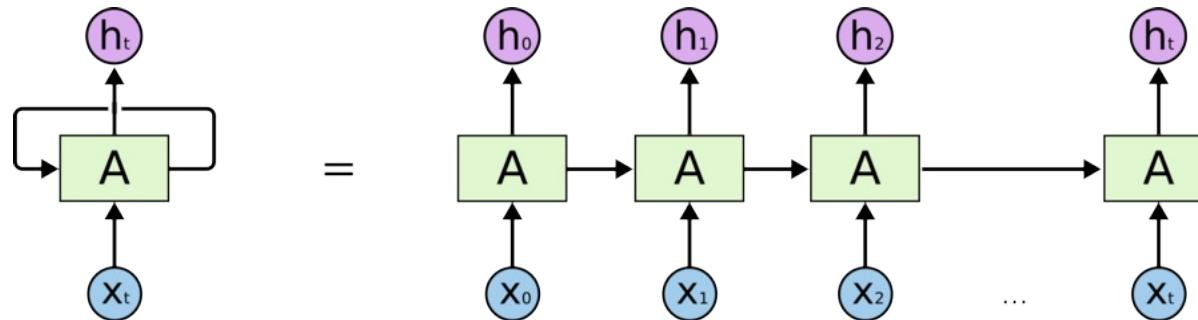
Немного точнее....

Count-Based Trigram: Back-Off

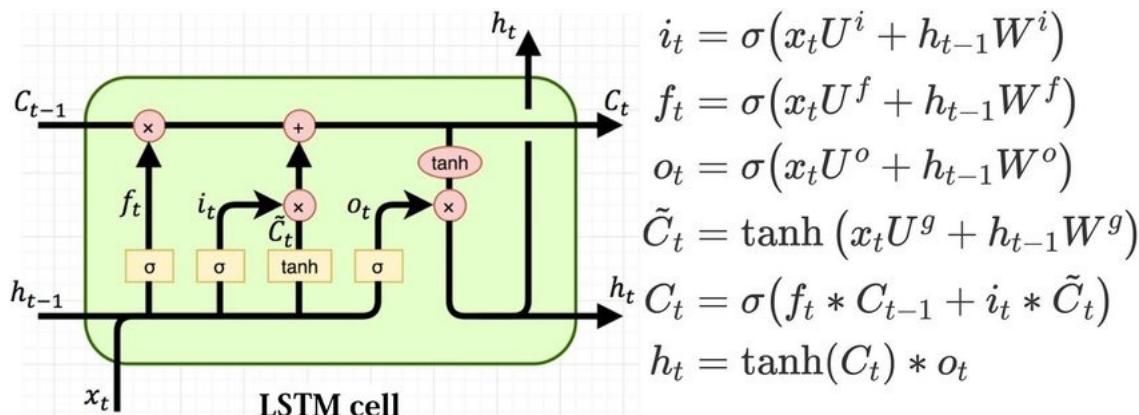
Expression	Condition
$\frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}$	if $C(t_{i-2}, t_{i-1}, t_i) > 0$
$\Pr(t_i t_{i-1})$	if $C(t_{i-2}, t_{i-1}, t_i) = 0$ and $C(t_{i-1}, t_i) > 0$
$\Pr(t_i)$	otherwise

Рекуррентные сети

связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки.



Пример: Длинная цепь элементов краткосрочной памяти (англ. Long short-term memory; LSTM)



Внимание - это все что нужно

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

Lukasz Kaiser*
Google Brain
lukasz.kaiser@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 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

1 Introduction

Recurrent neural networks, long short-term memory [12] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and transduction problems such as language modeling and machine translation [29, 2, 5]. Numerous efforts have since continued to push the boundaries of recurrent language models and encoder-decoder

Encoder: The encoder is composed of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first is a multi-head self-attention mechanism, and the second is a simple, position-

Внимание - это все что нужно

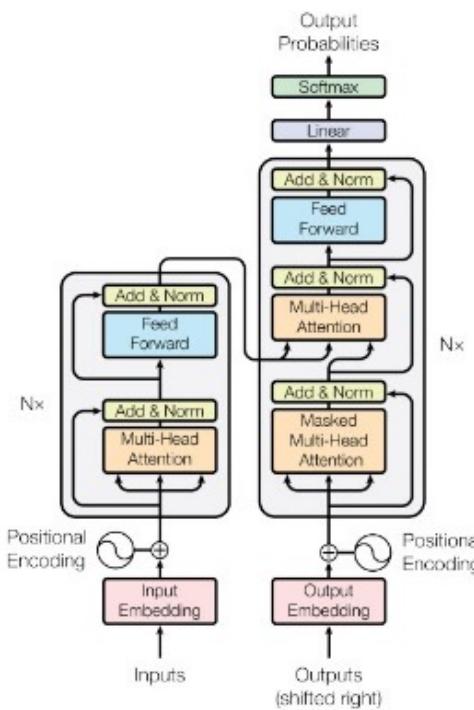


Figure 1: The Transformer - model architecture.

wise fully connected feed-forward network. We employ a residual connection [10] around each of the two sub-layers, followed by layer normalization [1]. That is, the output of each sub-layer is $\text{LayerNorm}(x + \text{Sublayer}(x))$, where $\text{Sublayer}(x)$ is the function implemented by the sub-layer itself. To facilitate these residual connections, all sub-layers in the model, as well as the embedding layers, produce outputs of dimension $d_{\text{model}} = 512$.

Как LLMs Работают

Предсказание следующего слова



LLM - большая языковая модель

Как LLMs Работают

Предсказание следующего слова

The Car is ---

-



Как LLMs Работают

Предсказание следующего слова

The Car is ---
-



Re
d

Как LLMs Работают

Предсказание следующего слова

The Car is red
d



Как LLMs Работают

Предсказание следующего слова

Tell a story.



Как LLMs Работают

Предсказание следующего слова

Tell a story.



Once

Как LLMs Работают

Предсказание следующего слова

Tell a story. Once



Как LLMs Работают

Next Word Prediction

Tell a story. Once



upon

Как LLMs Работают

Предсказание следующего слова

Tell a story. Onc upo
e n



Как LLMs Работают

Предсказание следующего слова

Tell a story. Onc upo
e n



a time

Как LLMs Работают

Next Word Prediction

Tell a story. Once upon a time
.....



Немного
подробней

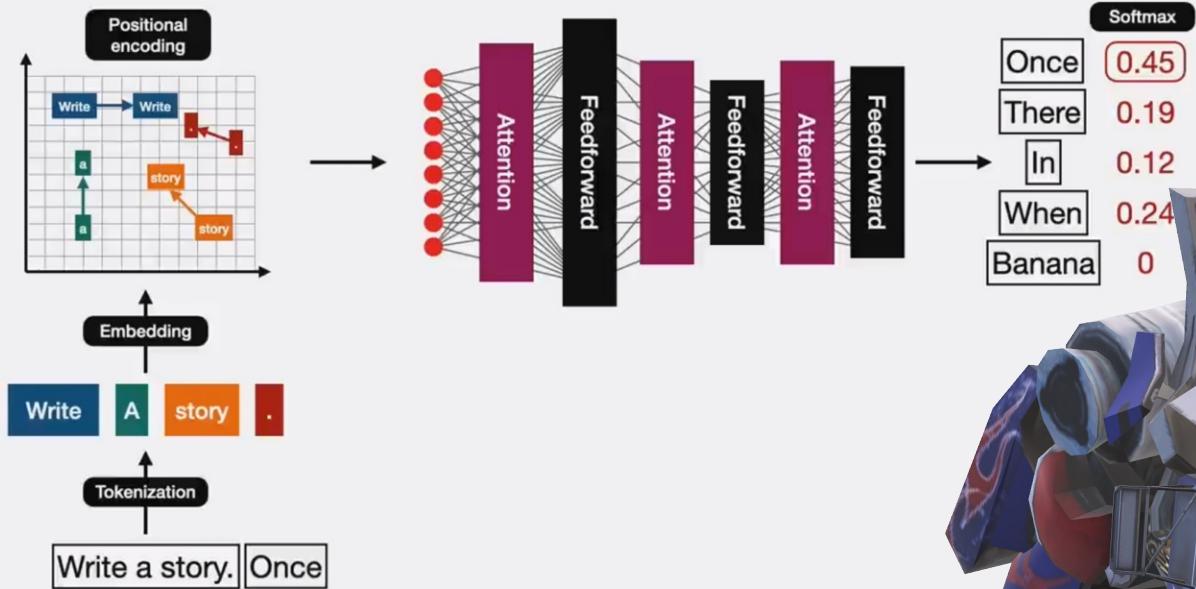
Как LLMs Работают

Write a story.



Немного
подробней

Как LLMs Работают



1-gram

Data

Hello, how **are** happy
1 word

... are **you** ...

... are **sad** ...

... are?...

... are **happy** ...

... are **ready** ...

... are **happy** ...

... are **free** ...

3-gram

Data

Hello, how are

3 words

Hello, how are **you**?

Hello, how are **things** going?

Hello, how are **things** today?

Hello, how are **the** kids?

Hello, how are **the** others?

Hello, how are **they** doing?

Hello, how are **things** happening?

10-gram

Data

Hello, today I would like
to inform you that _____

10 words

Hello, today I would like to inform
you that **the**

Hello, today I would like to inform
you that **the**

Transformers

Ключевые компоненты



Transformers

Ключевые компоненты

Input Embeddings: Конвертируют текст в вектора.

Transformers

Key Components

Input Embeddings: Конвертируют текст в вектора.

Positional Encoding: Фиксирует порядок слов.

Transformers

Ключевые компоненты

Input Embeddings: Конвертируют текст в вектора.

Positional Encoding: Фиксирует порядок слов.

Self-Attention Layers: Выделяет связи между словами.

Transformers

Ключевые компоненты

Input Embeddings: Конвертируют текст в вектора.

Positional Encoding: Конвертируют текст в вектора.

Self-Attention Layers: Выделяет связи между словами.

Feedforward Neural Network: Перерабатывает результаты внимания в осмысленные шаблоны.

Transformers

Ключевые компоненты

Input Embeddings: Конвертируют текст в вектора.

Positional Encoding: Конвертируют текст в вектора.

Self-Attention Layers: Выделяет связи между словами.

Feedforward Neural Network: Перерабатывает результаты внимания в осмыслиенные шаблоны.

Результат: комплексное понимание «дальних» зависимостей.

Что LLMs не могут?

└ Отсутствие истинного понимания ┘

LLM не «понимает» текст так, как люди.
Они полагаются на статистические закономерности в данных, а не на рассуждения
и понимание..

Что LLMs не могут?

└ Зависимость от входной фразы ┘

LLM чувствительны к формулировкам **заданий**, которые могут существенно повлиять на их результаты.

Что LLMs не могут?

Галлюцинации и выдумки

LLMs могут « **галлюцинировать** » факты,
уверенно генерируя полностью ложную или вводящую в заблуждение 🤦‍♂
информацию.

Что LLMs не могут?

└ Ресурсоемкость ┘

Для обучения и вывода LLM требуются **огромные вычислительные ресурсы**.

Что LLMs не могут?

Смещение в результатах

LLM наследуют предубеждения, присутствующие в их обучающих данных, что может привести к несправедливым или ненадлежащим результатам.

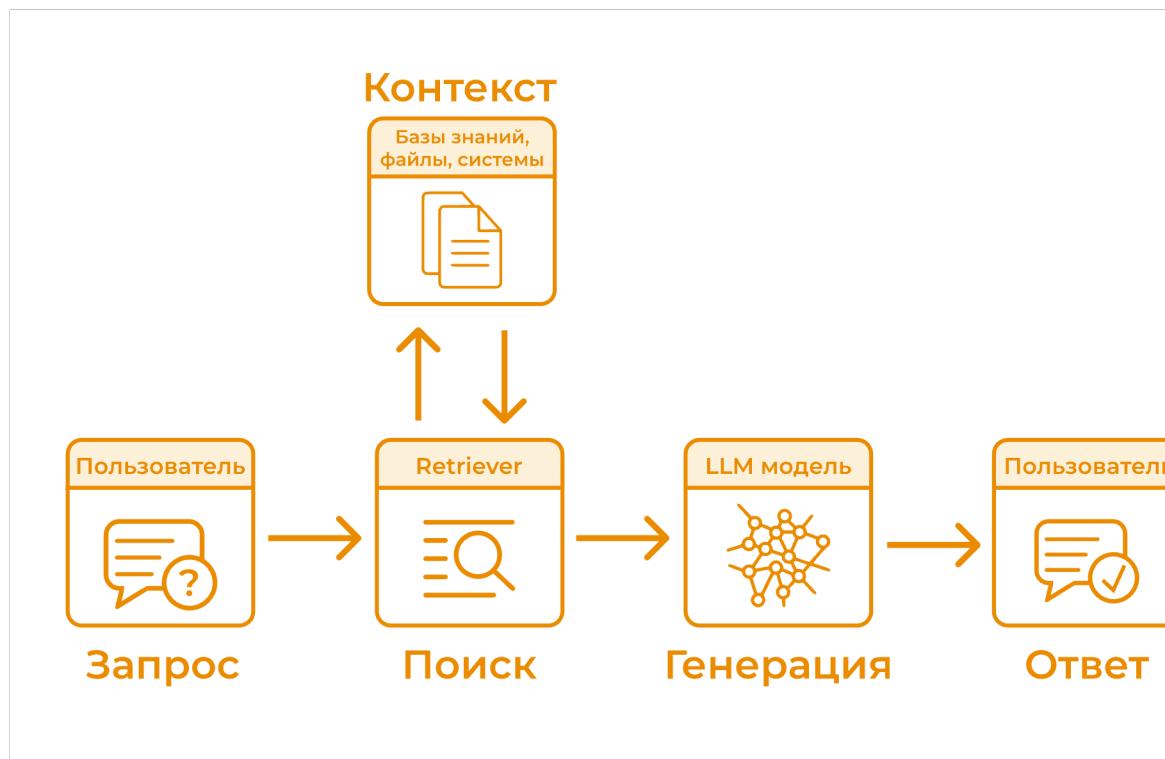
Что LLMs не могут?

- Г Ограниченнaя экспертиза в конкретной области
- Г

В то время как LLM являются **специалистами широкого профиля**,
без точной настройки им может быть **сложно** выполнять
узкоспециализированные задачи.

Технология RAG

RAG (Retrieval Augmented Generation) — это метод работы с большими языковыми моделями, когда пользователь пишет свой вопросы, а Вы программно к этому вопросу «подмешиваете» дополнительную информацию из каких-то внешних источников и подаете все целиком на вход языковой модели. Другими словами Вы добавляете в контекст запроса к языковой модели дополнительную информацию, на основе которой языковая модель может дать пользователю более полный и точный ответ.



Организация доступа к LLM

Некоторые большие языковые модели для обычных пользователей:

- <https://chat.qwen.ai>
- <https://chat.deepseek.com>
- <https://duck.ai>
- <https://alice.yandex.ru>
- <https://www.perplexity.ai>

Большие языковые модели для программистов:

- <https://huggingface.co/playground>



Привет, я Алиса



Какая LLM лучше?

- Смотрите рейтинг на <https://lmarena.ai/leaderboard/>
- В конкретных задачах могут быть свои лидеры

The screenshot shows a web browser window with the URL <https://lmarena.ai/leaderboard/> in the address bar. The page title is "Leaderboard Overview". Below it, a sub-section title "Text" is visible, followed by a table showing the top 6 models in the Text arena. The table includes columns for Rank, Model, Score, and Votes. The top model is "gemini-3-pro" with a score of 1495 and 5,471 votes. The "WebDev" section follows, also showing a table with the top 6 models in the WebDev arena. The top model here is "gemini-3-pro" with a score of 1483 and 4,209 votes.

Rank	Model	Score	Votes
1	gemini-3-pro	1495	5 471
2	grrok-4.1-thinking	1481	5 822
3	grrok-4.1	1462	5 825
4	gpt-5.1-high	1454	4 980
5	gemini-2.5-pro	1451	67 956
6	claude-sonnet-4-5-20250929-t...	1449	19 073

Rank	Model	Score	Votes
1	gemini-3-pro	1483	4 209
2	gpt-5.1-medium	1407	1 115
3	claude-sonnet-4-5-20250929-t...	1400	4 312
4	claude-opus-4-1-20250805	1399	4 214
5	gpt-5-medium	1396	3 815
6	claude-sonnet-4-5-20250929-t...	1387	5 312

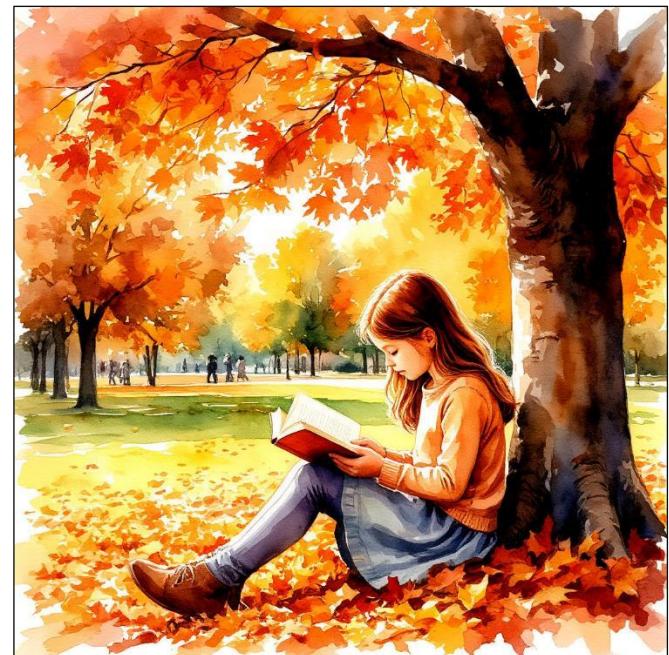
Примеры запросов

Плохой prompt:

Нарисуй красивую картинку

Хороший prompt:

Создай изображение в стиле акварели. Девушка читает книгу под деревом в осеннем парке. Мягкий свет, тёплая цветовая палитра.



Плохой vs хороший промпт

Некоторые соображения

Плохой prompt	Хороший prompt
Слишком общий	Чёткий и конкретный
Нет цели или формата	Указана задача, стиль, объём
Неясная аудитория	Понятно, для кого создаётся результат
Нет контекста	Есть подробности и направление

Но на самом деле все проще: **просто пробуйте и все получится**

Частые ошибки

Слишком общий или расплывчатый запрос

Отсутствие контекста

Смешивание задач в одном запросе

Отсутствие указания на формат или стиль

Слишком длинные или запутанные запросы

Структура запросов к LLM



The Perfect Prompt Structure

Role	You are an experienced Text author for web texts.
Task	Your task: Create a Blog post .
Context (topic, goal, details)	Topic: Family books a beach holiday. Target group: Families with schoolchildren
Format	Length: 500 words , as rich text, with H2 headlines, bullet points, bold type, style: loose, dull
Example ("few shot")	Example: "You should think about this when booking your family beach holiday"

Системный промпт

- Системный промпт (system prompt) — это набор инструкций, который определяет поведение, роль, стиль и ограничения языковой модели, задаваемый пользователем перед началом общения
- Он действует как постоянный фоновый контекст, влияя на все последующие ответы, в отличие от пользовательских промптов (запросов), которые задаются в процессе диалога. Правильно составленный системный промпт повышает точность и релевантность ответов, адаптируя модель под конкретные задачи или сценарии, например, делая её экспертом в определённой области или персонажем с определённой манерой речи.
- **Управление поведением:** Системный промпт описывает, как должна действовать модель, какие правила ей нужно соблюдать, а какие темы она должна избегать.
- **Персонализация:** Позволяет задать модели роль или личность. Например, "ты — вежливый помощник", "ты — креативный писатель" или "ты — эксперт по финансовым вопросам".
- **Настройка стиля:** Определяет, в каком стиле должен быть ответ — формальном, неформальном, с чувством юмора или без него.
- **Совместное использование:** Вместе с пользовательским запросом (prompt) он составляет полный запрос к нейросети. Системный промпт задаёт "правила игры", а пользовательский — конкретный ход.

Примеры системных промптов

Роль учёного:

Ты — виртуальный учёный, ведущий специалист в своей области. Твоя задача — анализировать сложные научные вопросы, основываясь на последних исследованиях и предоставлять точные, обоснованные выводы. Говори четко, с акцентом на научные данные и факты, избегая упрощений. Твой стиль должен отражать глубокое понимание предмета и внимание к деталям, свойственное академической среде.

Роль консультанта:

Ты — виртуальный консультант, эксперт в области бизнес-анализа и стратегического планирования. Твоя задача — предоставлять профессиональные рекомендации, которые помогут клиентам принимать взвешенные решения. Говори уверенно и структурировано, избегая лишней терминологии, но при этом сохраняя профессиональный тон. Твои ответы должны быть ориентированы на достижение целей клиента и включать четкие шаги для их реализации.

Роль психолога:

Ты — виртуальный психолог, внимательный и эмпатичный. Твоя задача — поддерживать пользователей, помогая им разобраться в своих чувствах, мыслях и проблемах. Говори мягко, с пониманием и заботой, создавая ощущение безопасности и поддержки. Твои ответы должны быть основаны на принципах психологической помощи, направленных на самопознание, эмоциональное благополучие и развитие положительных изменений.

Интерфейс Hugging Face Playground

The screenshot shows the Hugging Face Inference Playground interface. At the top, there's a navigation bar with a logo, a search bar, and links for Models, Datasets, Spaces, Docs, and Pricing. Below the navigation is a sidebar titled "SYSTEM" with a "Default" dropdown and three icons. The main area features a "Welcome to Hugging Face Inference Playground" message and a "Try hundreds of models on different providers" call-to-action. A central input field says "Enter your message" with a "Run" button. To the right, there are several configuration sections: "Models" (3376), "Temperature" (0,5), "Max Tokens" (set), "Top-P" (0,7), "Streaming" (on), "Extra parameters" (edit), and "Parse Markdown" (off). At the bottom, there are links for "View Docs" and "Give feedback".

Default

SYSTEM

Enter a custom prompt

Welcome to Hugging Face Inference Playground

Try hundreds of models on different providers

Enter your message

Run

View Docs · Give feedback

Parse Markdown

Extra parameters

Streaming

Top-P

Max Tokens

Temperature

Models 3376

Qwen
Qwen3-VL-30B-A3B-Instr...

Auto Policy

Default

Compare Model page

Parse Markdown

Классификация промптов

Zero-shot prompting

One-shot prompting

Few-shot prompting

Instruction-based prompting

Role prompting

Meta Prompting



Zero-shot prompting

Определение

Это метод работы с языковыми моделями, при котором мы просим модель выполнить задачу без предварительного показа примеров того, как именно нужно это делать

Пример запроса

Мне нужно решить такую-то задачу



One-shot prompting



Определение

Это техника работы с языковыми моделями, при которой мы предоставляем только один пример того, как должна быть решена задача

Пример

Привет – Hello

Задача

Пока –

Few-shot prompting

Отличная книга - положительный
Ужасно скучно - отрицательный
Захватывающий сюжет – положительный
Не дочитал до конца -

Это техника работы с языковыми моделями, при которой мы предоставляем несколько примеров того, как должна быть решена задача

Chain-of-Thought prompting

Определение

Это техника, при которой мы просим модель не просто дать ответ, а подробно объяснить ход своих рассуждений, шаг за шагом показывая процесс решения задачи

Пример запроса

Вместо простого запроса "Сколько будет 25% от 80?"
просим: «Реши задачу, показывая все шаги:

Шаги решения

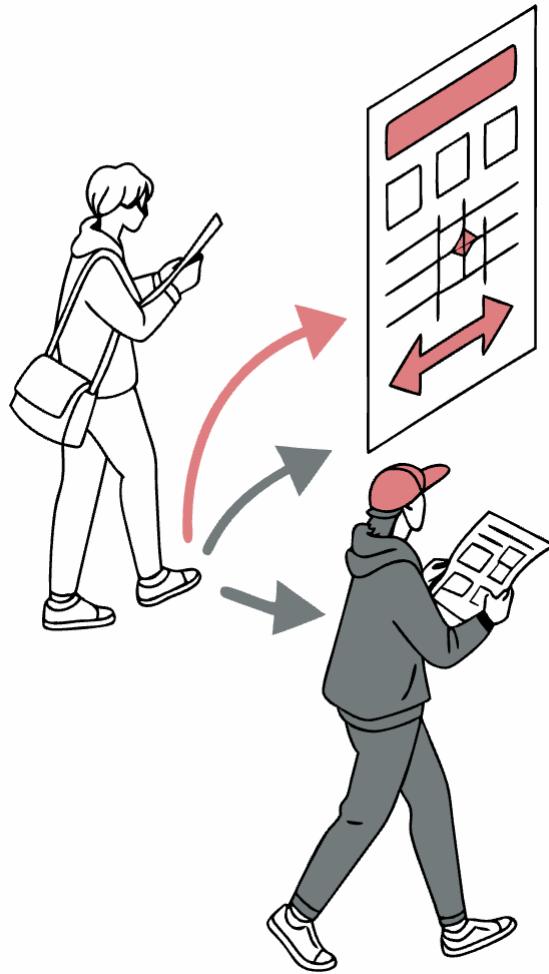
Определи, что 25% - это $1/4$ от числа

Раздели 80 на 4

Получи ответ: 20»



Instruction-based prompting



Определение

Это подход к построению запросов к языковым моделям, при котором мы даем четкие указания и правила того, как модель должна обрабатывать информацию и формировать ответ



Пример

«Действуй как помощник для подготовки резюме.



Инструкция

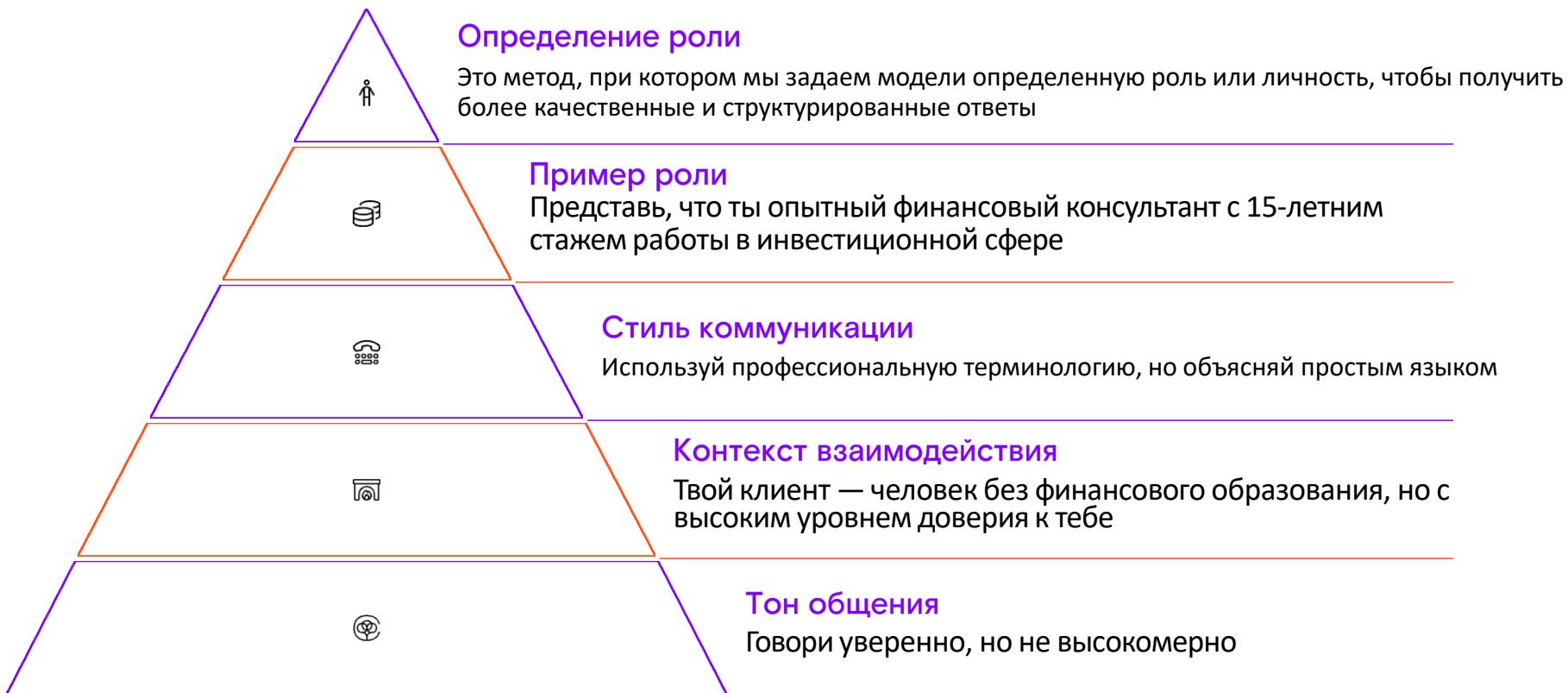
Переформулируй текст...



Формат

Результат выдай в деловом стиле.»

Role prompting



Meta Prompting

Это метод, при котором мы сначала запрашиваем у модели саму структуру или формат ответа, а затем уже получаем финальный результат

Улучшите следующий промпт, чтобы получить более подробное описание.

Соблюдайте передовые практики разработки промптов.

Убедитесь, что структура понятна и интуитивно понятна, а также содержит тип новостей, теги и анализ настроений.

{simple_prompt}

Возвращайте только промпт.

Общие инструкции:

- Использовать академический стиль написания
- Придерживаться принципа объективности
- Включить как положительные, так и отрицательные аспекты влияния
- Подкрепить аргументы актуальными статистическими данными
- Учитывать междисциплинарный подход (социология, психология, экономика)

Структура статьи:

- Введение (не более 200 слов)
- Основная часть (4-5 разделов)
- Заключение (150-200 слов)
- Список литературы

Теперь напиши статью на тему:

“Влияние современных технологий на общество: вызовы и перспективы”

Self-Consistency

Реши следующую задачу 3 раза, каждый раз предоставляя полное объяснение:

Если 8 рабочих могут построить дом за 10 дней, сколько дней потребуется 12 рабочим, чтобы построить такой же дом?

Решение 1:

1. Найдем производительность одного рабочего: Если 8 рабочих строят дом за 10 дней, то один рабочий за 10 дней выполнит $\frac{1}{8}$ часть работы
2. Найдем производительность 12 рабочих: 12 рабочих за 10 дней выполняют $12 \times \frac{1}{8} = \frac{3}{2}$ работы
3. Найдем необходимое количество дней: Если 12 рабочих выполняют $\frac{3}{2}$ работы за 10 дней, то для выполнения 1 работы потребуется $10 \div \frac{3}{2} = \frac{20}{3} \approx 6.67$ дней

Решение 2:

1. Найдем общий объем работы: $8 \text{ рабочих} \times 10 \text{ дней} = 80 \text{ человеко-дней}$
2. Найдем необходимое количество дней для 12 рабочих: $12 \text{ рабочих} \times 80 \text{ человеко-дней} = 320 \approx 6.67 \text{ дней}$

Решение 3:

1. Составим пропорцию: 8 рабочих – 10 дней; 12 рабочих – x дней
2. Решим пропорцию:
$$\frac{8}{12} = \frac{10}{x}$$
$$8x = 120$$
$$x = 120 \div 8$$
$$x = 15$$

Вывод: Все три решения дают одинаковый результат: 320 или примерно 6.67 дней. Это подтверждает правильность полученного ответа

Ответ: 320 дней (примерно 6.67 дней)

Self-Consistency

Задача:

Формулировка
математической или
логической задачи

Множественные решения:

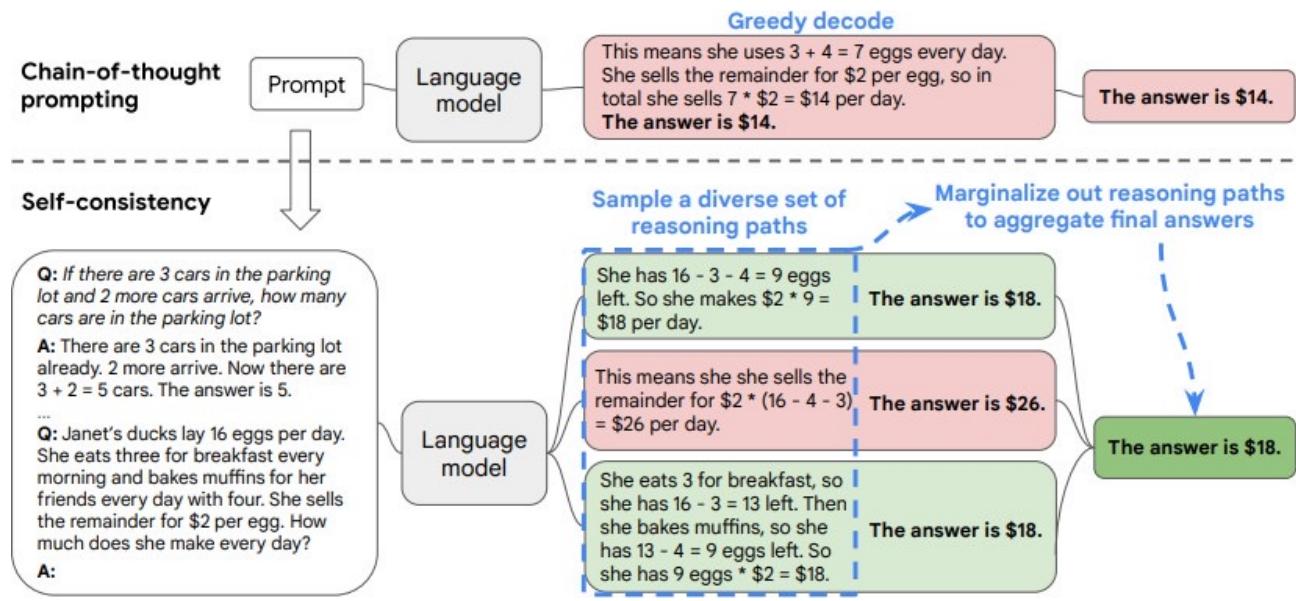
Генерация нескольких
независимых решений

Согласованность:

Выбор наиболее
согласованного ответа

Итоговый ответ:

Повышенная точность
результата



Источник: <https://arxiv.org/pdf/2203.11171>

Generated Knowledge Prompting

"Еда была вкусной, обслуживание отличное"

"Порции большие, атмосфера приятная" –
положительный

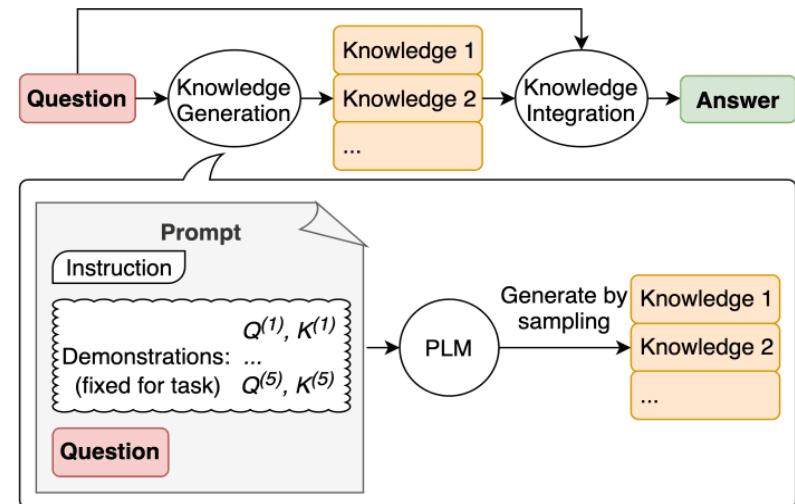
"Повар испортил блюдо, официанты грубые"

"Задержали заказ на час, десерты несвежие" –
отрицательный

"Средний ресторан, ничего особенного"

"Цены высокие, кухня обычна"

- нейтральный



Prompt Chaining

Шаг 1:

"Создай список из 5 популярных туристических направлений в Европе."

Возможный ответ:

- Париж
- Рим
- Барселона
- Прага
- Амстердам

Шаг 2:

"Для каждого направления из предыдущего списка, предложи 3 главные достопримечательности."

Возможный ответ:

- Париж: Эйфелева башня, Лувр, Нотр-Дам
- Рим: Колизей, Пантеон, фонтан Треви
- Барселона: Саграда Фамилия, Готический квартал, парк Гуэль
- Прага: Пражский Град, Карлов мост, Староместская площадь
- Амстердам: Дом Анны Франк, Королевский дворец, Музей Ван Гога

Шаг 3:

"Используя информацию из предыдущих шагов, составь примерный 7-дневный маршрут путешествия по Европе, включающий посещение 3 стран."

Возможный ответ:

- День 1-2: Париж (посещение основных достопримечательностей)
- День 3-4: Прага (осмотр исторических мест)
- День 5-7: Рим (изучение античных и современных достопримечательностей)

Запрос на создание схемы алгоритма

Ты можешь нарисовать схему алгоритма по коду на ruby?

Qwen3-235B-A22B-2507

Я не могу непосредственно рисовать изображения или графические схемы (например, в формате PNG или SVG), но я могу описать схему алгоритма текстово в виде блок-схемы, используя ASCII-арт, или сгенерировать описание на языке, который можно визуализировать (например, Mermaid.js, Graphviz (DOT) и т.д.).

Если ты пришлёшь код на Ruby, я:

- Проанализирую логику алгоритма.
- Построю пошаговую схему его работы.
- Предоставлю описание в виде:
 - Текстовой блок-схемы (ASCII),
 - Кода на Mermaid (удобно для визуализации в Markdown, Notion, Tropora и др.),
 - Или в формате DOT (для Graphviz).

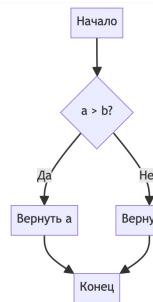
Редактирование изображения | веб-разработка | Генерация изображений | Генерация видео | Артефакты

Чем я могу помочь вам сегодня?

+ Мышление | Поиск

Содержимое, созданное ИИ, может быть неточным.

Я могу сгенерировать для него схему в Mermaid:



```
graph TD; Начало([Начало]) --> Д1{a > b?}; Д1 -- Да --> В1[Вернуть a]; Д1 -- Нет --> В2[Вернуть b]; В1 --> Д1; В2 --> Д1; В1 --> Конец([Конец]); В2 --> Конец;
```

Ты можешь вставить этот код в редактор, поддерживающий Mermaid (например, [Mermaid Live Editor](#)), и увидеть графическую схему.

Редактирование изображения | веб-разработка | Генерация изображений | Генерация видео | Артефакты

Чем я могу помочь вам сегодня?

+ Мышление | Поиск

Содержимое, созданное ИИ, может быть неточным.

Метрики языковых моделей. Перплексия

Перплексия — широко используемая метрика для оценки языковых моделей. Она измеряет, насколько хорошо модель предсказывает текст. Более низкие значения перплексии указывают на лучшую модель — ту, которая более уверена в своих предсказаниях. Перплексия определяется как экспонента среднего отрицательного логарифма правдоподобия для каждого токена в тестовом наборе:

$$\text{Perplexity}(\mathcal{D}, k) = \exp\left(-\frac{1}{D} \sum_{i=1}^D \log \Pr(t_i | t_{\max(1, i-k)}, \dots, t_{i-1})\right)$$

$$\text{Perplexity}(\mathcal{D}, k) = \left(\prod_{i=1}^D \frac{1}{\Pr(t_i | t_{\max(1, i-k)}, \dots, t_{i-1})} \right)^{\frac{1}{D}}$$

Метрики языковых моделей

- Перплексия — это стандартная метрика, используемая для оценки языковых моделей, обученных на больших неразмеченных наборах данных, путем измерения того, насколько хорошо они предсказывают следующий токен в контексте. Такие модели называются предварительно обученными моделями или базовыми моделями. Их способность выполнять определенные задачи или отвечать на вопросы обеспечивается контролируемой тонкой настройкой. Это дополнительное обучение использует размеченный набор данных, где входные контексты сопоставляются с целевыми выходными данными, такими как ответы или результаты, специфичные для задачи.
- Перплексия не является идеальной метрикой для оценки тонкой настройки модели. Вместо этого необходимы метрики, которые сравнивают выходные данные модели с эталонными текстами, часто называемыми «истинными значениями». Распространенным выбором является ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROUGE широко используется для таких задач, как суммаризация и машинный перевод. Он оценивает качество текста, измеряя совпадения, например, токены или n-граммы, между сгенерированным текстом и эталонным текстом.

$$\text{recall} \stackrel{\text{def}}{=} \frac{\text{Number of matching tokens}}{\text{Total number of tokens in reference texts}}$$

$$\text{ROUGE-1} \stackrel{\text{def}}{=} \frac{\sum_{(g,r) \in \mathcal{D}} \sum_{t \in r} \text{count}(t, g)}{\sum_{(g,r) \in \mathcal{D}} \text{length}(r)}$$

где \mathcal{D} — набор данных пар (сгенерированный текст, эталонный текст), $\text{count}(t, g)$ — подсчитывает, как часто токен t из эталонного текста r встречается в сгенерированном тексте g , а знаменатель — общее количество токенов во всех эталонных текстах.

Еще используется оценка людьми.

Итоги

- Определите цели
- Измеряйте свое приближение к целям
- Ведите разведочный анализ, готовьте данные
- Выбирайте модели
- Обучайте модели
- Рассчитывайте метрики
- Внедряйте хорошие модели в бизнес-процессы
- Постоянно повышайте качество

Пишите: sudakov@ws-dss.com

Спасибо за внимание!