СӘТБАЕВ УНИВЕРСИТЕТІ

SATBAYEV UNIVERSITY

Основы искусственного интеллекта. Лекция 10. Продвинутые методы NLP. Секвенсные модели, RNN, LSTM, трансформеры

Преподаватель: **Қасенхан Арай Мейрамбайқызы, доктор PhD, ассоциированный профессор кафедры «Программная инженерия»**

a.kassenkhan@satbayev.university

Содержание

1. **Последовательные данные**
2. **RNN**
3. **Проблемы RNN**
4. **LSTM и GRU**
5. **Transformer**
6. **Self-Attention**
7. **BERT · GPT · T5**
8. **Hugging Face**

По завершению урока Вы будете знать:

1. **Почему порядок слов важен для NLP**
2. **Как RNN обрабатывает последовательность**
3. **Зачем появились LSTM и GRU**
4. **Почему Transformer заменил RNN во многих NLP-задачах**
5. **Как работает идея Self-Attention**
6. **Чем отличаются BERT, GPT и T5**
7. **Как используются готовые NLP-модели**

Почему текст — это последовательность?

Пример:

«Собака укусила человека»

≠

«Человек укусил собаку»

Те же слова → другой порядок → другой смысл.

Для NLP важны:

- **Порядок слов**
- **Контекст**
- **Связи между словами**
- **Долгосрочные зависимости**

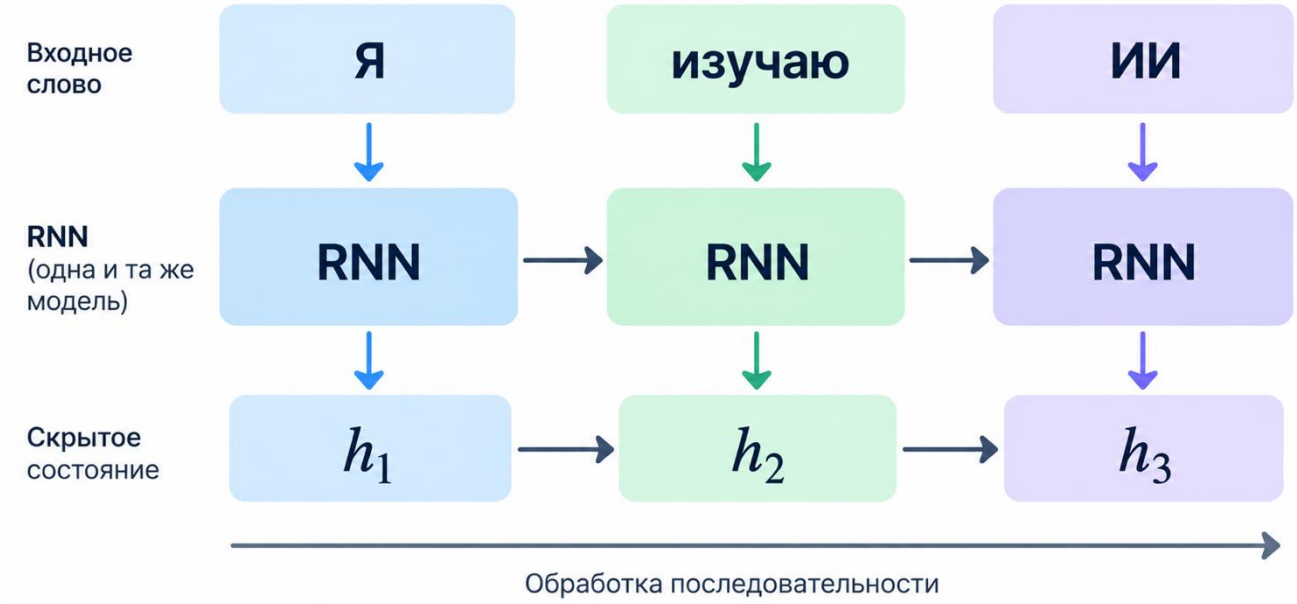
Текст — последовательность, в которой значение элемента зависит от контекста.

Зачем нужны модели последовательностей?



RNN: сеть с памятью

RNN (Recurrent Neural Network) — нейронная сеть, которая обрабатывает последовательность шаг за шагом и передаёт информацию о предыдущих шагах дальше.

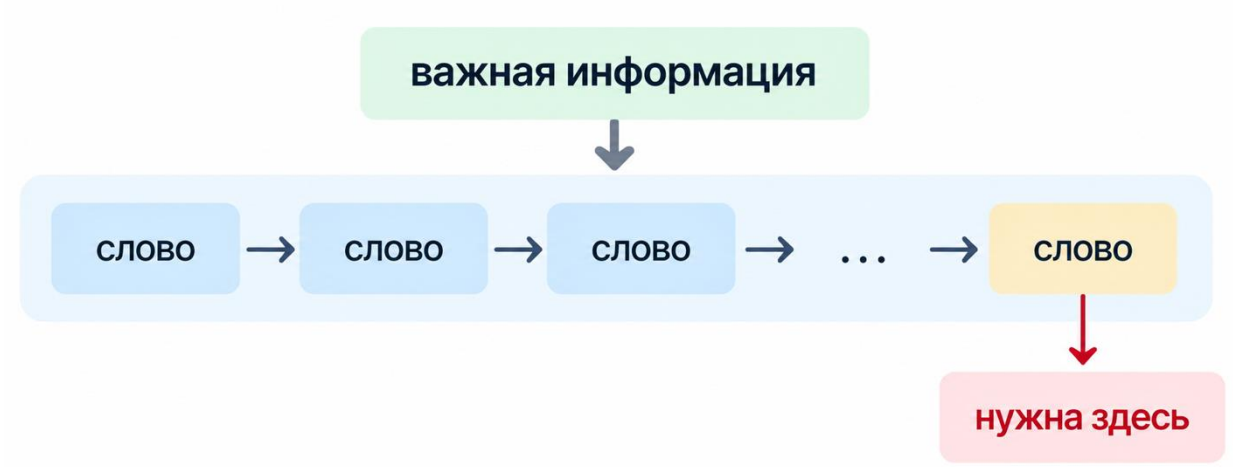


Hidden State = память о предыдущем контексте

Текущее состояние зависит от текущего входа и предыдущего состояния.

Проблема RNN: дальний контекст

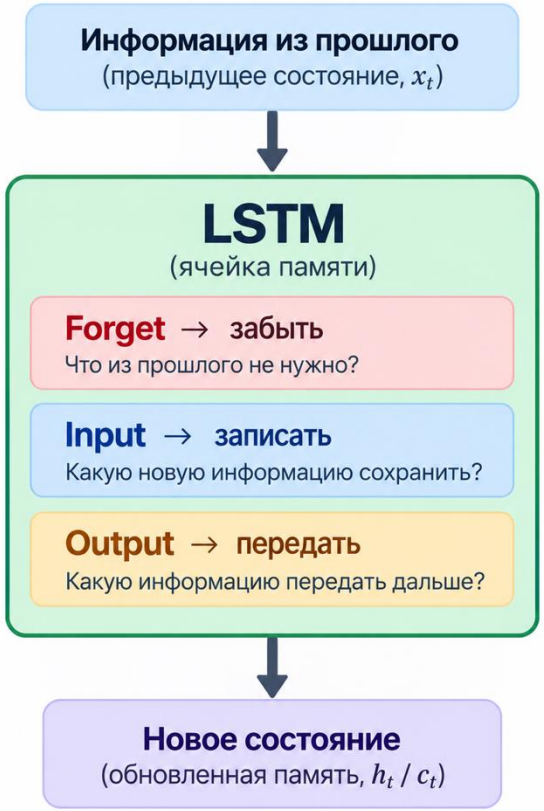
Например:
«Я вырос во Франции, много лет жил в Париже, учился и работал там, поэтому свободно говорю на …»
Чтобы предсказать:
«французском»
модели нужно сохранить информацию, которая появилась значительно раньше.



RNN трудно сохранять информацию через большое количество шагов.
Одна из причин — проблема исчезающего градиента при обучении.

LSTM: что нужно помнить, а что забыть?

LSTM (Long Short-Term Memory) — развитие RNN, предназначенное для лучшего сохранения долгосрочной информации.



Forget Gate → что забыть
Input Gate → что сохранить
Output Gate → что передать дальше

LSTM управляет потоком информации с помощью специальных gates.

GRU: более простая альтернатива LSTM

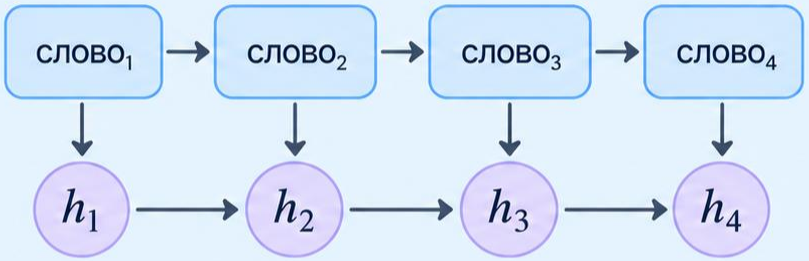
LSTM	GRU
Более сложная архитектура	Более простая
3 основных gate	2 основных gate
Больше параметров	Меньше параметров
Хорошо работает с долгими зависимостями	Часто обучается быстрее

GRU сохраняет основную идею управляемой памяти, но использует более простую архитектуру.

RNN → LSTM / GRU

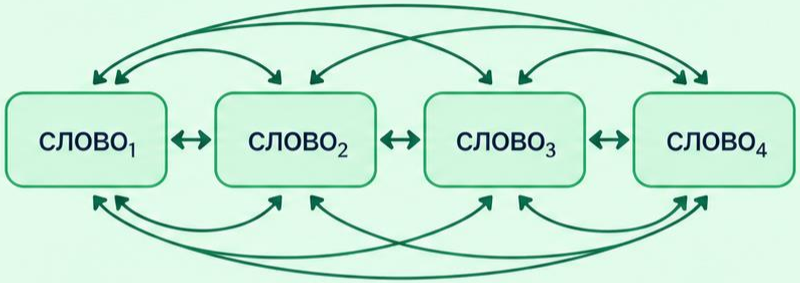
Почему понадобился Transformer?

RNN / LSTM



- Последовательная обработка
- Сложнее работать с дальними зависимостями

Transformer



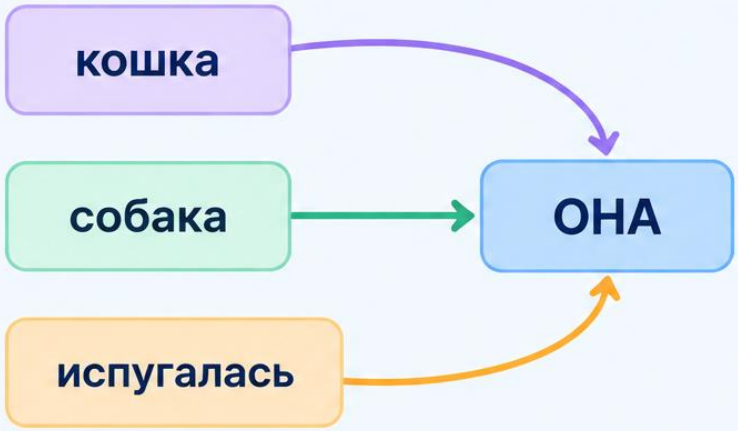
- Self-Attention
- Связи между разными позициями
- Параллельная обработка при обучении

Transformer отказался от рекуррентной обработки и сделал центральным механизм внимания.

Self-Attention: на что обратить внимание?

Пример

«Кошка увидела собаку, потому что **она** испугалась.»



Self-Attention

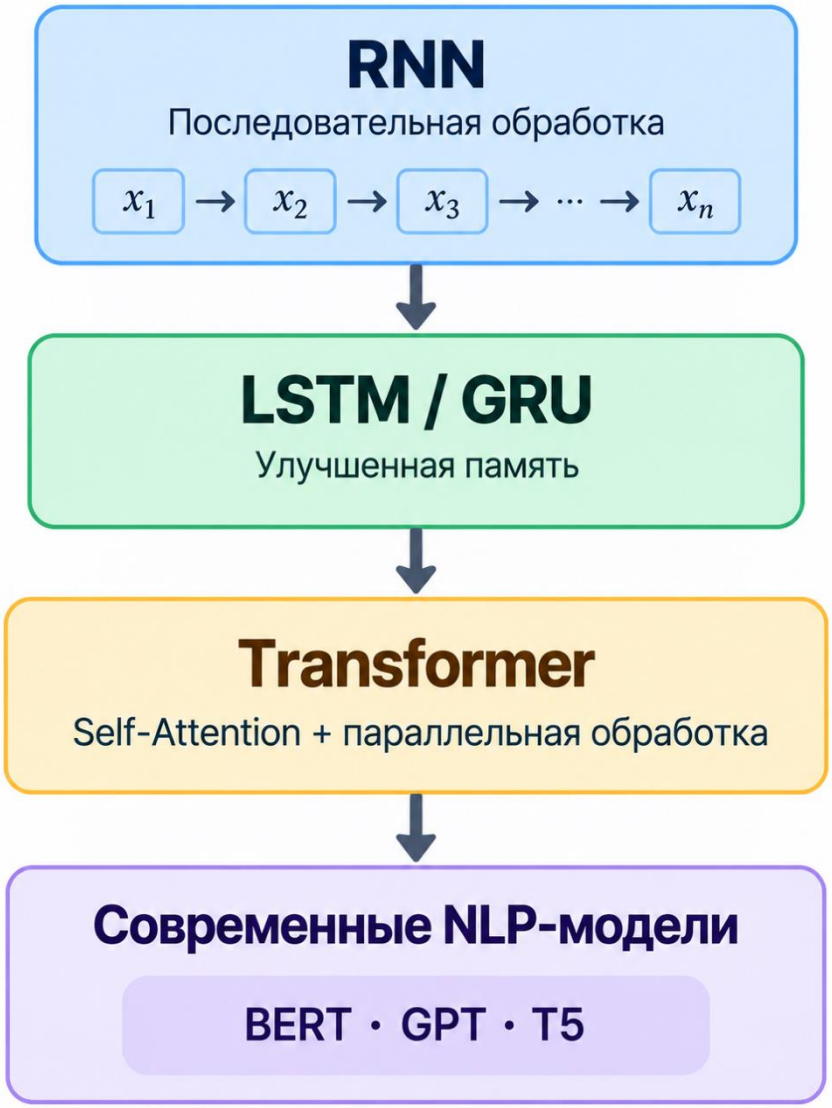
позволяет модели определить, какие слова важны друг для друга в данном контексте.

Q Query → что ищем

K Key → с чем сравниваем

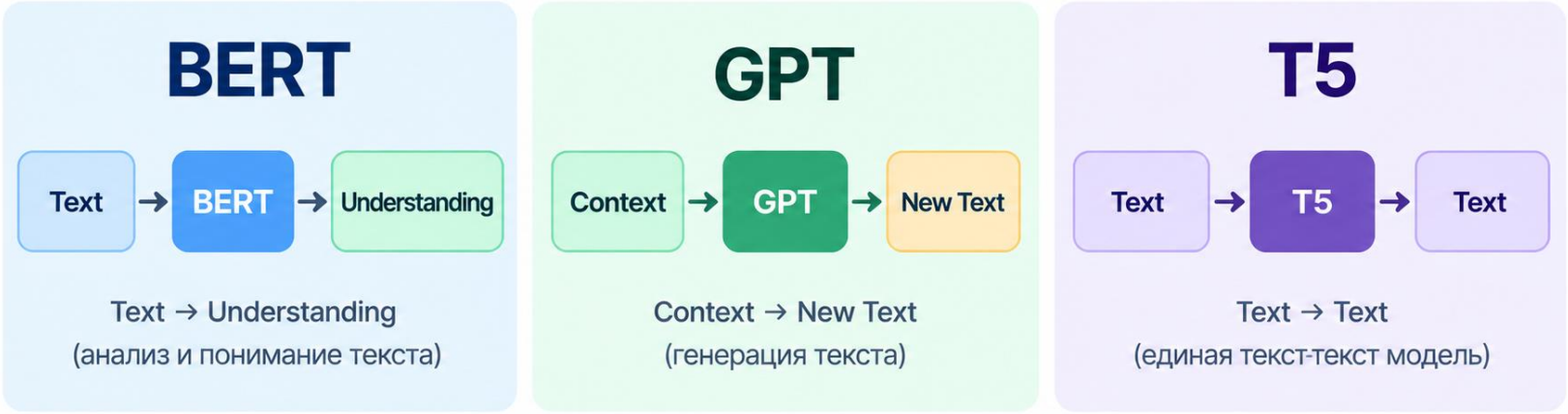
V Value → какую информацию используем

Эволюция моделей для текста



BERT, GPT и T5: в чём разница?

	BERT	GPT	T5
Архитектурная идея	Encoder	Decoder	Encoder–Decoder
Основной акцент	Понимание текста	Генерация	Text-to-Text
Типичные задачи	Classification, NER	Generation, dialogue	Translation, summarization



RNN → LSTM → Transformer

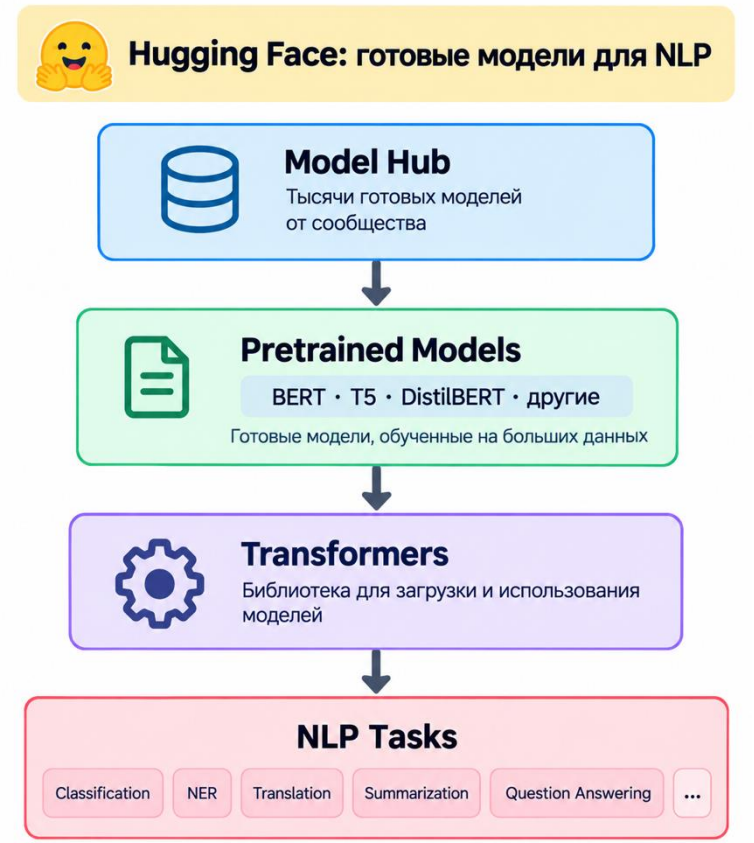
	RNN	LSTM / GRU	Transformer
Обработка	Последовательная	Последовательная	Параллелизуемая при обучении
Дальний контекст	Ограниченно	Лучше	Self-Attention
Скорость обучения	Ограничена последовательностью	Ограничена последовательностью	Хорошо масштабируется
Сложность	Низкая	Средняя	Высокая
Современный NLP	Реже	Используется	Доминирующий подход

От архитектуры к готовой модели



Во многих задачах не нужно обучать языковую модель с нуля — можно использовать предобученную модель.

Hugging Face




```
from transformers import pipeline

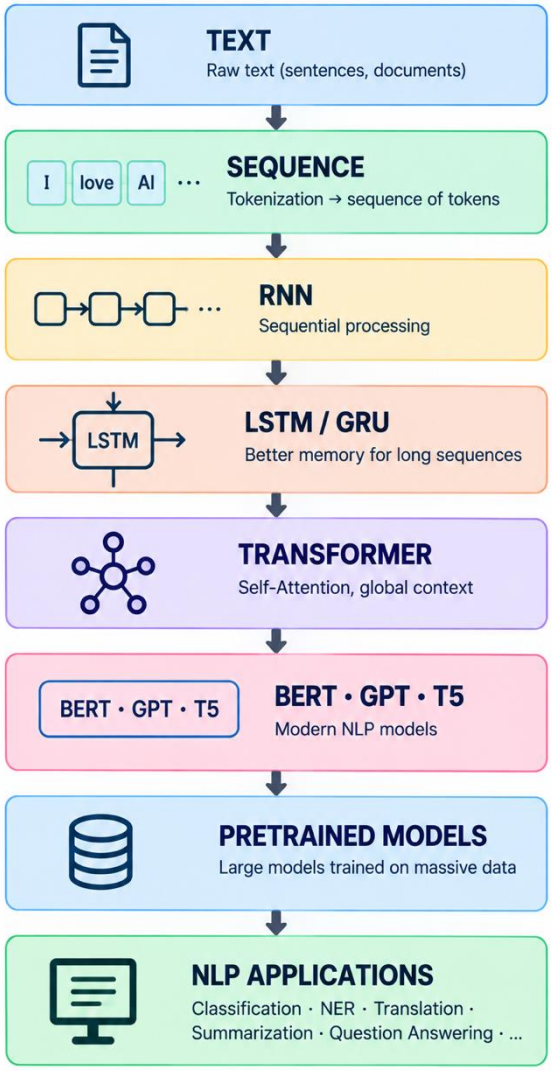
classifier = pipeline("sentiment-analysis")

classifier("I love artificial intelligence!")
```

Результат условно:
POSITIVE

pipeline позволяет использовать предобученную модель через простой интерфейс.

Заключение



RNN → память о предыдущих шагах
LSTM / GRU → лучше сохраняют долгосрочную информацию
Transformer → Self-Attention и масштабирование
Pretrained Models → готовые знания можно использовать для новых задач

Современный NLP прошёл путь от последовательной обработки текста к большим предобученным моделям на основе Transformer.