

Основы практического использования нейронных сетей.

Лекция 9. Трансформеры.

Дмитрий Буряк.
к.ф.-м.н.
dyb04@yandex.ru

Задачи анализа естественного языка

□ Natural Language Processing (NLP) – применение методов распознавания образов в словах, предложениях, текстах.

□ Виды задач NLP:

- Классификация текста (text classification)
- Фильтрация контента (content filtering)
- Классификация отзывов (sentiment analysis)
- Моделирование языка (language modeling)
- Перевод (translation)

ChatGPT

- ❑ Chat Generative Pre-Trained Transformer
- ❑ Дообученная модель GPT-3
- ❑ 175 млрд параметров
- ❑ Выпущен в ноябре 2022

Content Creation

Call for action lines for twitter View	Crafting Customer Testimonials for your product or service View
Create personalised chatbot conversations View	Creating a viral reddit headline View
Domain name generator View	Explain concept as a poem View

Categories

Assistant	Automation	Clothing & Apparel	Content Creation	Cooking		
Creativity & Experiments	Customer Support	Data	Design	Exam & Competition		
Fitness	Games	Grammar	Healthcare	Home	Jailbreak / Tricks / Hacks	
Language	Learning	Legal	Machine Learning	Marketing	Music	Opinion
Product	Programming	Prompt Writing	Search Engine	SEO	Startups	Writing

Assistant

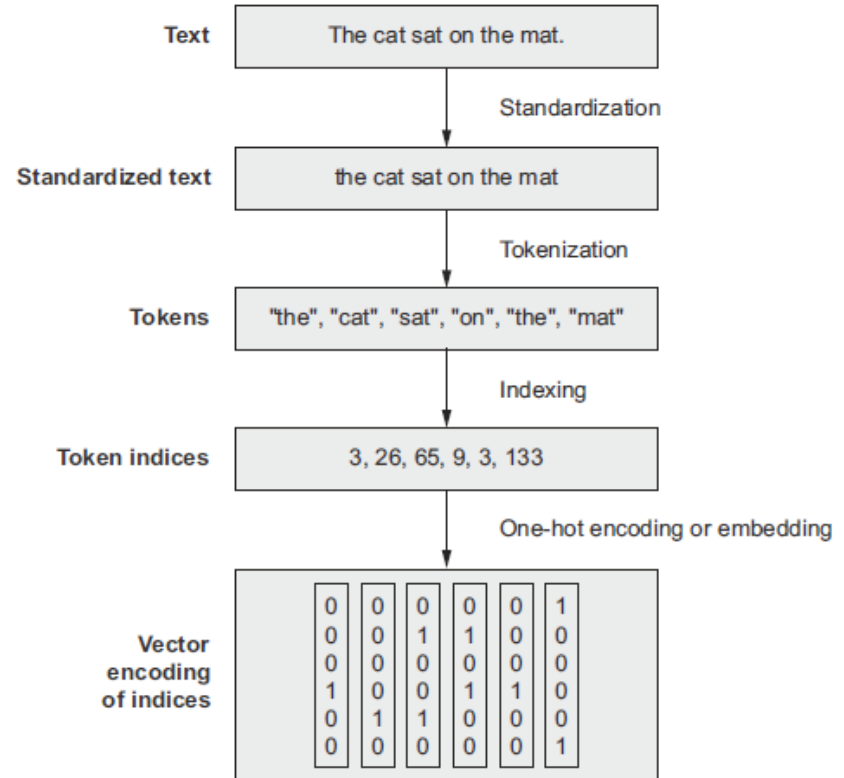
CEO Duties - Agenda for all-hands meeting View	CEO Duties - Company values to build a culture View
College suggestions for PhD in a field View	Create a shopping list View

Customer Support

Answer Shopify support queries View	Compose Complex Spreadsheet Formulas View
Create a list of pain points for target user base that can be addressed View	Create email automation sequences that will nurture leads and convert them into customers View

Предобработка текстовых данных

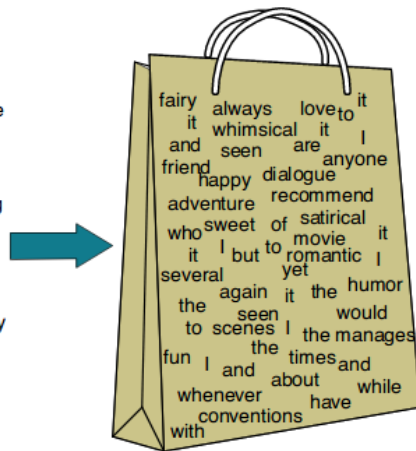
- ❑ Нормализация: приведение к прописным символам, удаление знаков пунктуации.
- ❑ Выделение токенов: символов, слов, групп слов (N-gram).
- ❑ Векторизация: преобразование токена в вектор (словарь → индекс слова в словаре → бинарный вектор/эмбединг).



Представление текста

- ❑ Текст – множество слов (N-gram) – bag-of-words.
 - сети прямого распространения, перцептроны.
- ❑ Текст – последовательность слов
 - Рекуррентные сети (LSTM, GRU)
 - Трансформеры

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!



it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

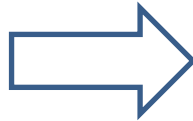
Представление типа bag-of-words

<https://koushik1102.medium.com/nlp-bag-of-words-and-tf-idf-explained-fd1f49dce7c4>

Векторизация. Дескрипторы слов.

❑ Бинарные вектора (on-hot):

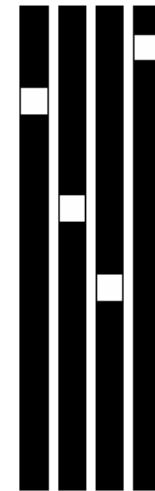
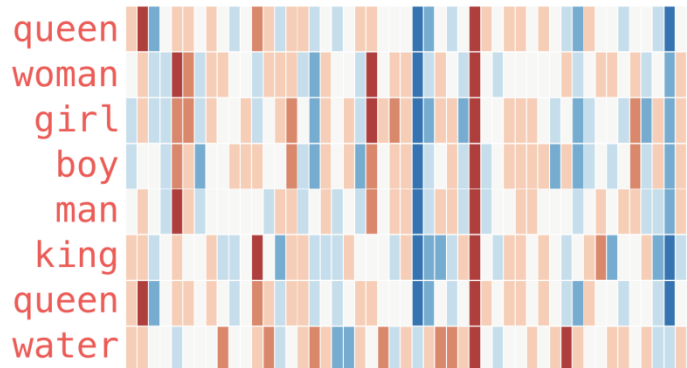
- высокая размерность
- ортогональны
- разрежены.



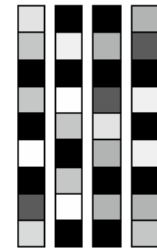
Не отражают свойств
естественного языка

❑ Формирование дескрипторов слов (эмбединг)

- вектора для близких по смыслу слов должны быть ближе, чем те, которые имеют разное значение
- получены в результате обучения
- word2vec
- GLoVe



One-hot word vectors:
- Sparse
- High-dimensional
- Hardcoded



Word embeddings:
- Dense
- Lower-dimensional
- Learned from data

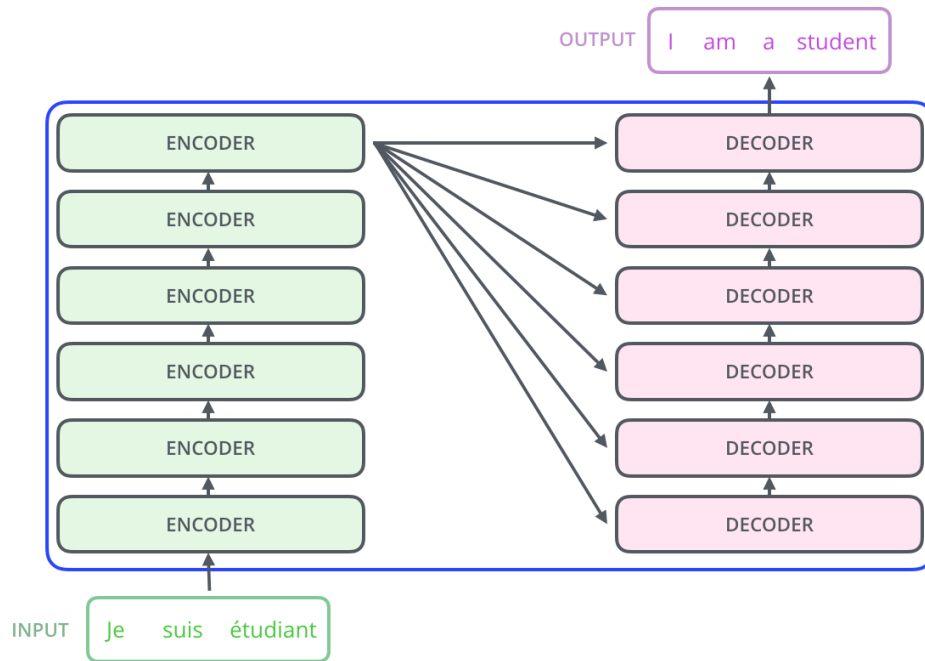
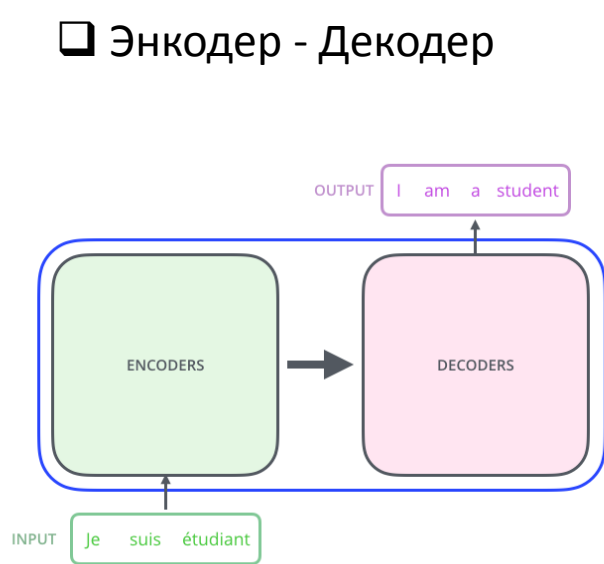
Трансформеры для обработки ТЕКСТОВ

- ❑ Vaswani et al. Attention is all you need, 2017
- ❑ Сеть прямого распространения, частично связанная, несколько параллельных ветвей, без сверточных слоев.
- ❑ Изначально предназначена для NLP, сейчас адаптирована для других задач, например, анализ изображений.



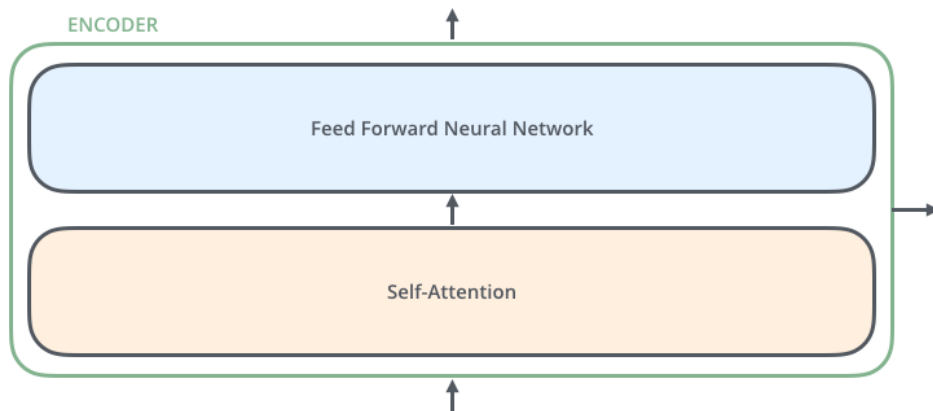
Структура трансформера

□ Энкодер - Декодер



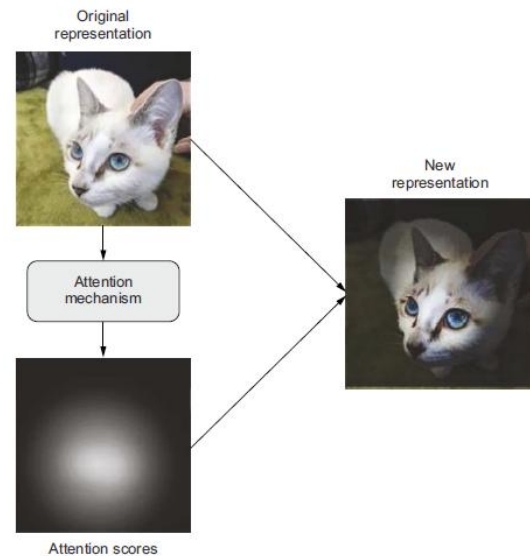
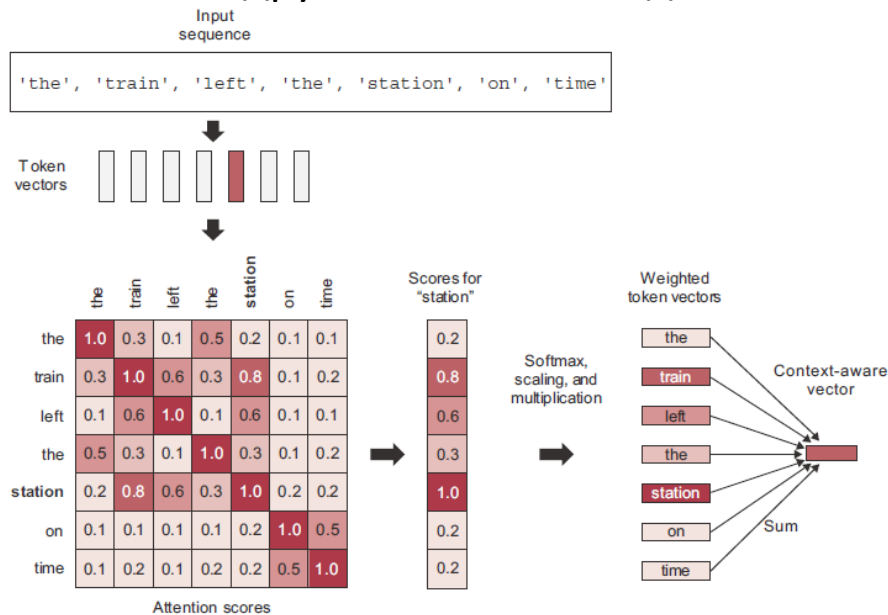
Энкодер

- ❑ Слой внимания: оценить взаимосвязь слов во входящей последовательности.
- ❑ Сеть прямого распространения для отображения результата слоя внимания



Механизм «Внимания»

- ❑ Слова могут иметь разное значение в зависимости от контекста
- ❑ Выделение признаков, зависящих от контекста.
- ❑ Цель – сформировать новое признаковое представление в зависимости от других слов в последовательности.

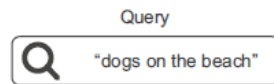


Идея реализации механизма «внимания» в глубоком обучении

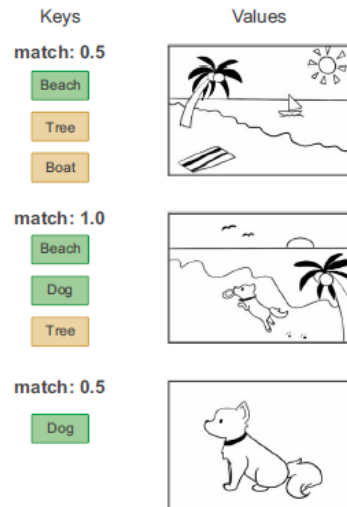
Модель “Query-Key-Value”

- ❑ Запрос (query) сравнивается с известными элементами данных (key) → коэффициент схожести.
- ❑ Для каждого известного элемента сопоставлен вектор значений (value).
- ❑ Результат сумма элементов значений, взвешенных коэффициентами схожести.

```
outputs = sum(values * pairwise_scores(query, keys))
```

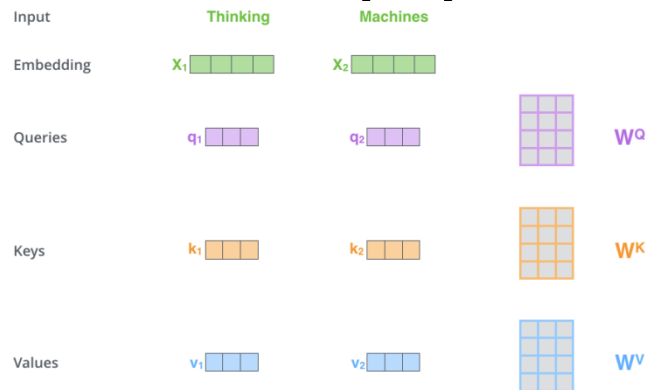


Применение модели “query-key-value” для ранжирования изображений по их релевантности текстовому запросу

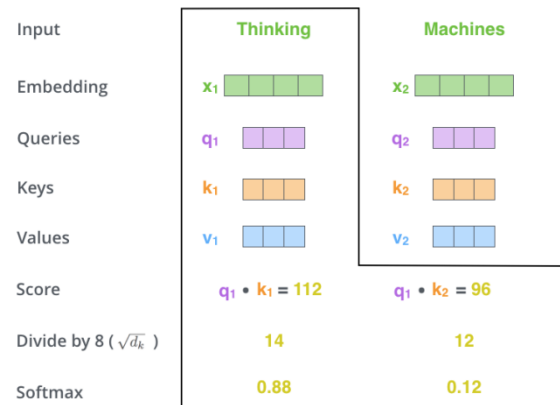


Пример “Query-Key-Value” (1)

□ Элементы запрос, ключ и значение – результат умножения входного эмбединга на соответствующую матрицу

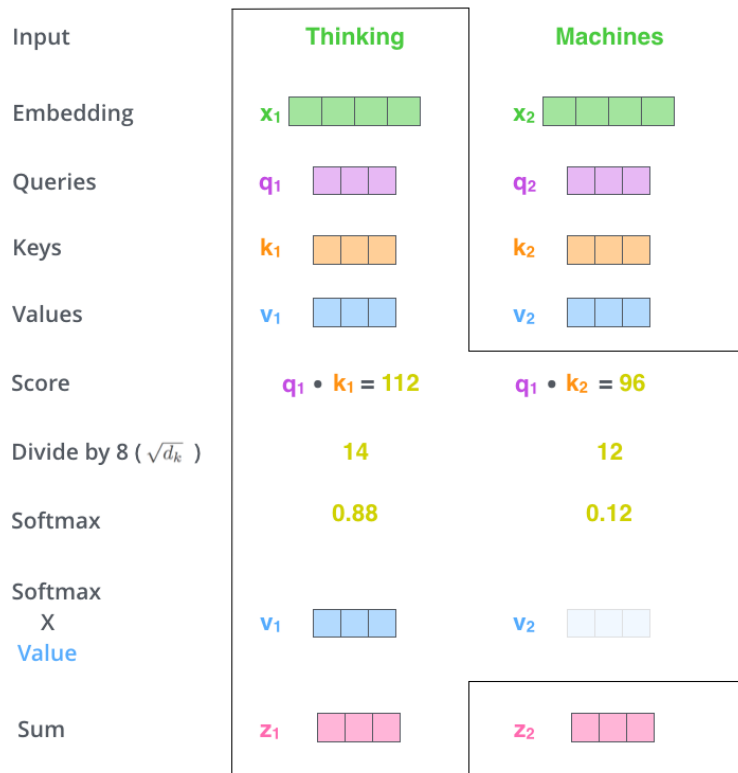


□ Вычисление коэффициента схожести и взвешивающих коэффициентов (деление на 8 – для данного примера)



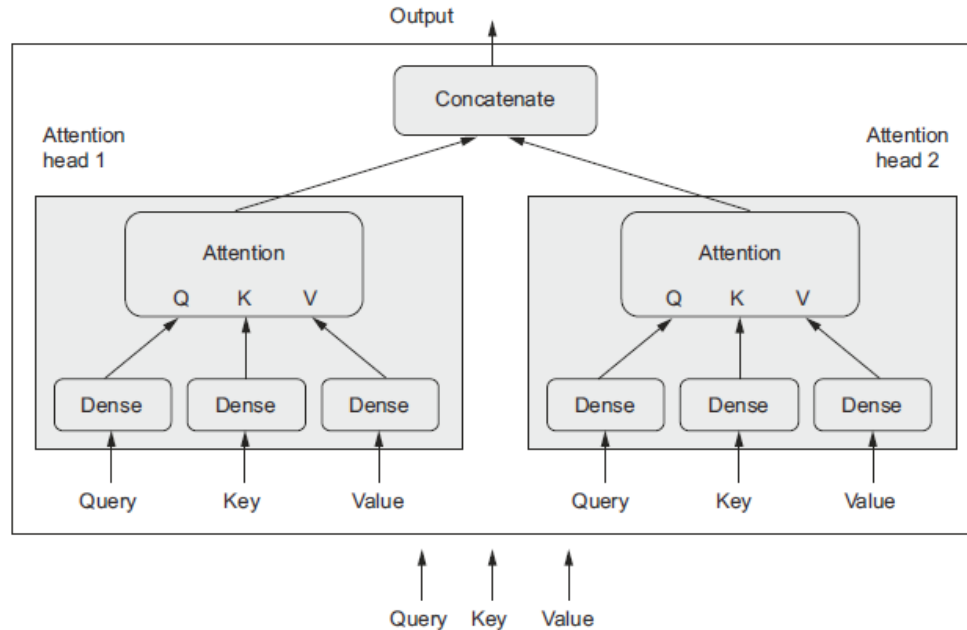
Пример “Query-Key-Value” (2)

□ Получение представления с учетом внимания для входного слова



Множественное внимание

- Формирование нескольких представлений (параллельно) для входной последовательности с последующей конкатенацией

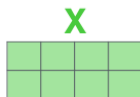


Пример множественного внимания

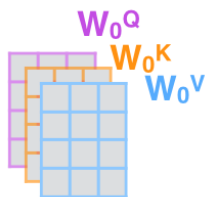
1) This is our input sentence*

Thinking
Machines

2) We embed each word*



3) Split into 8 heads. We multiply X or R with weight matrices



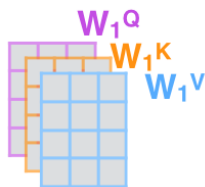
4) Calculate attention using the resulting Q/K/V matrices



5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer



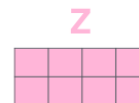
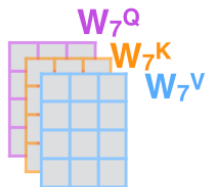
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



...

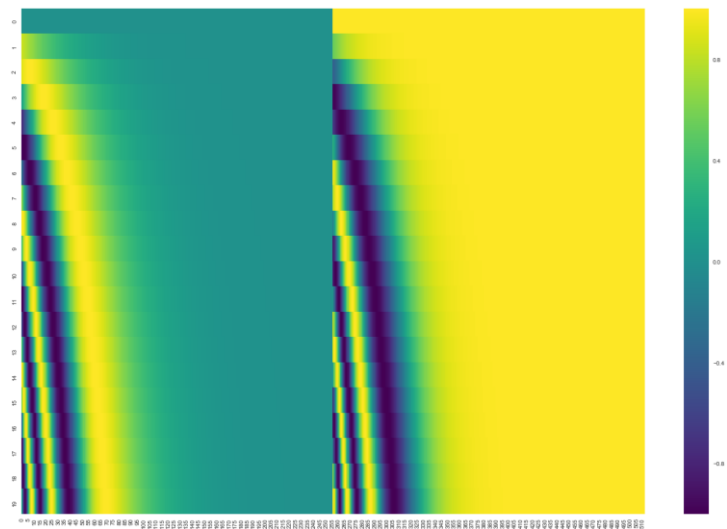
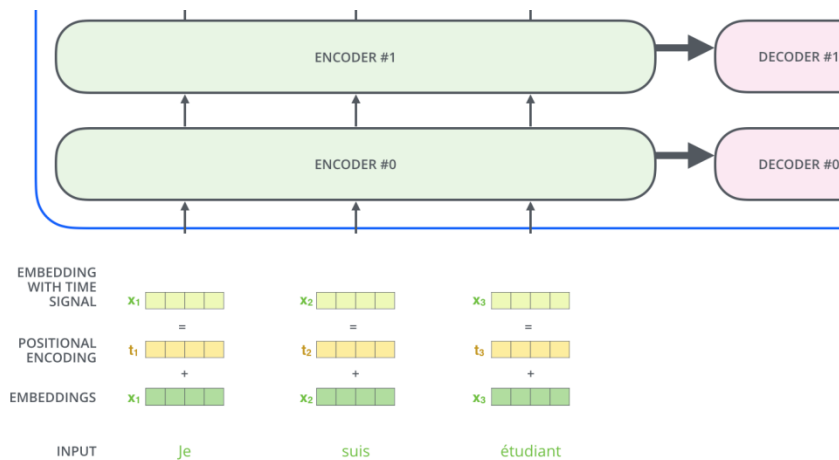
...

...



Позиционное кодирование

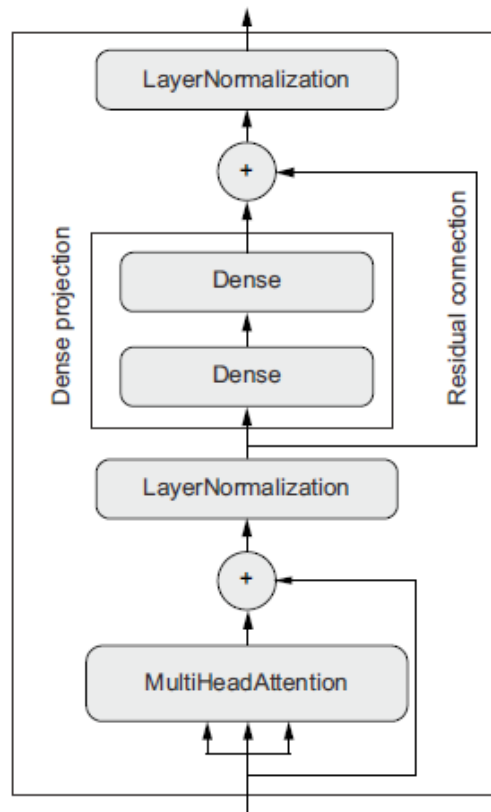
- ❑ Необходимо учитывать порядок слов в последовательности.
- ❑ К вектору эмбединг слова добавляется вектор, определяющий его позицию в предложении.



Пример позиционного кодирования для 20 слов. Длина вектора - 512

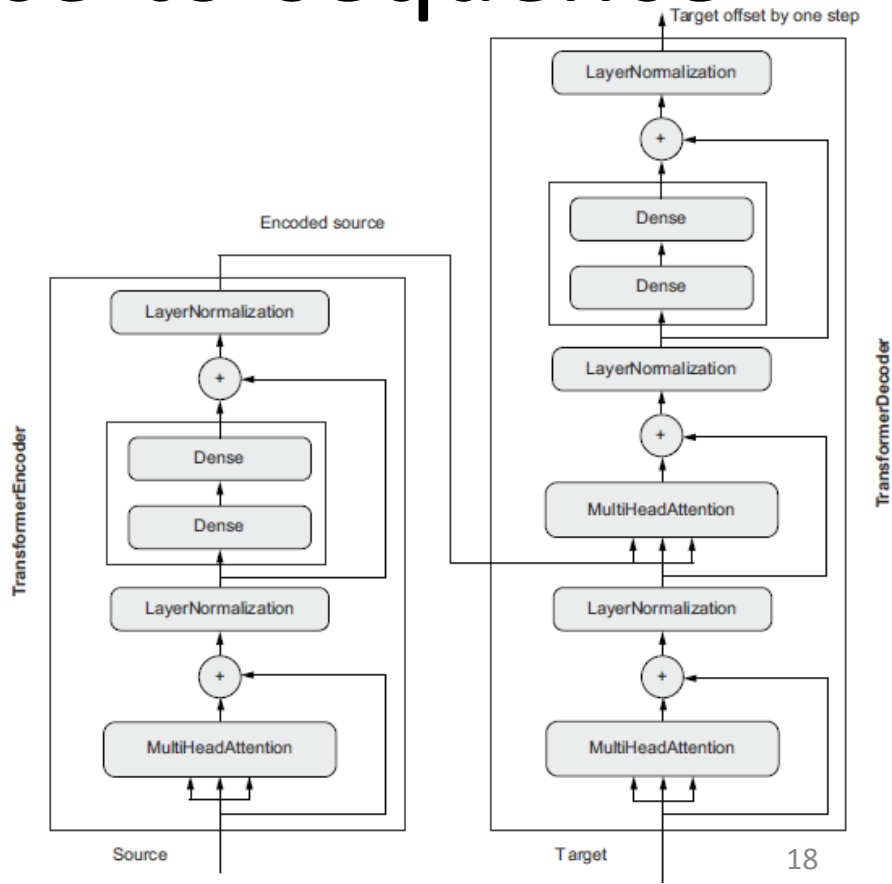
Архитектура энкодера

- ❑ Энкодер + классификатор = решение для классификации текстов.
- ❑ Энкодер + Декодер = модель для задач последовательность – последовательность (sequence-to-sequence)
 - перевод
 - составление абстракта для текста
 - ответ на вопрос
 - генерация текста.



Модель sequence-to-sequence

- Декодер предсказывает выходное слово на основании его взаимосвязи со словами в исходной последовательности:
 - выходное слово – запрос
 - исходная последовательность – ключ, значение



Генерация выходного слоя

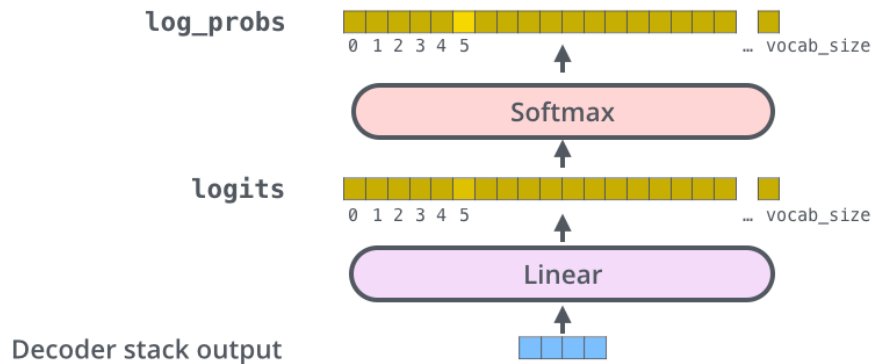
- Линейный слой + softmax = вектор (размер словаря)

Which word in our vocabulary
is associated with this index?

am

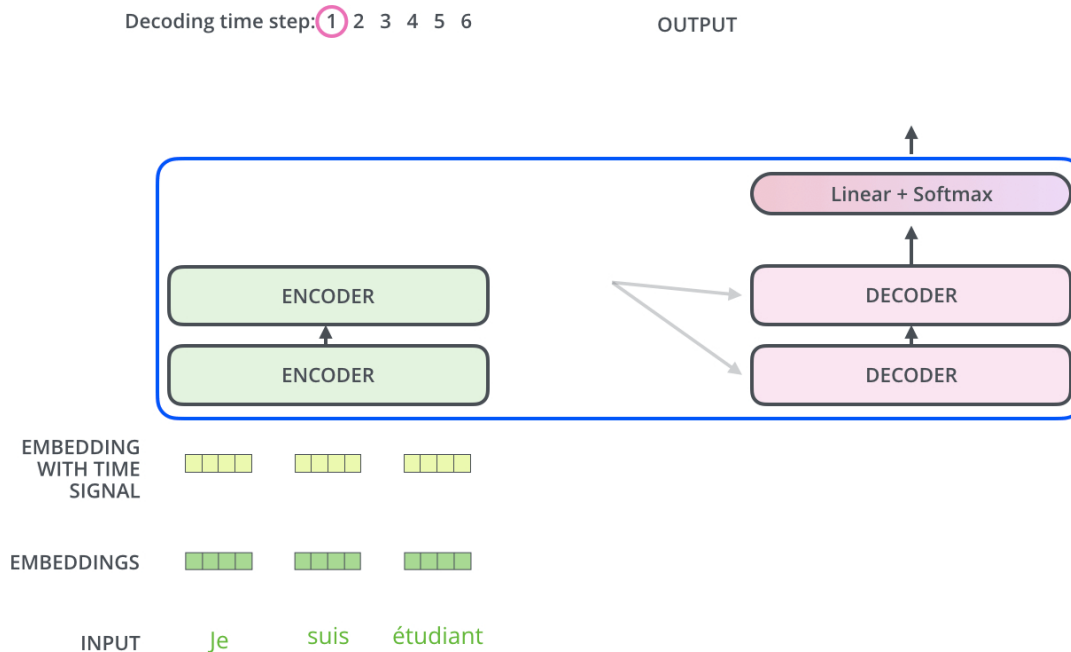
Get the index of the cell
with the highest value
(argmax)

5



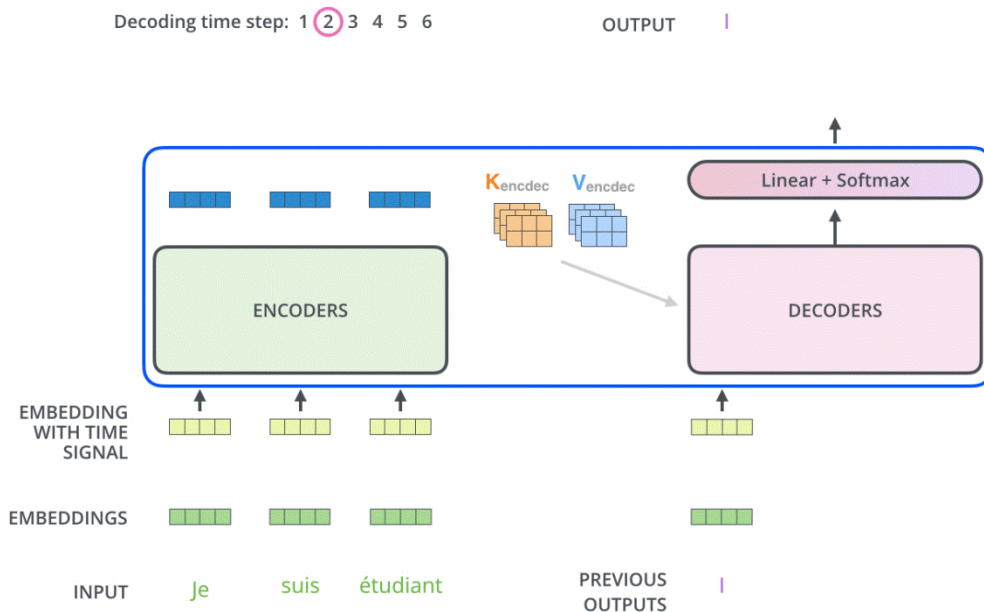
Пример sequence-to-sequence(1)

- ❑ Кодирование входной последовательности, получение первого выходного слова.



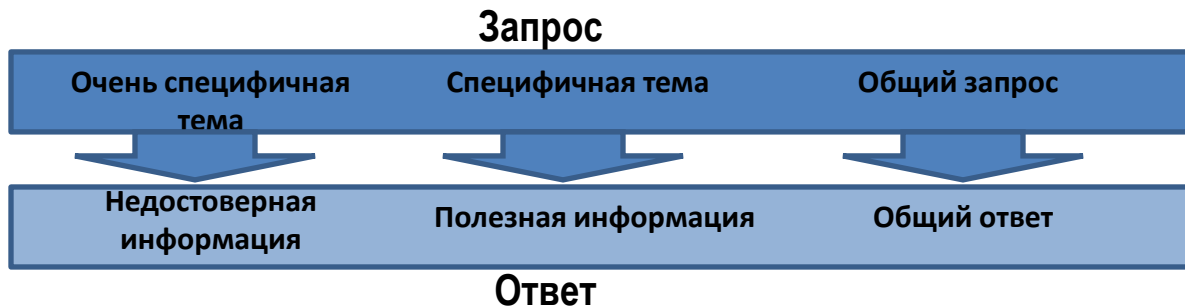
Пример sequence-to-sequence(2)

- ❑ Вызов декодера с текущим сгенерированным фрагментом пока не появится флаг окончания генерируемой последовательности.



ChatGPT. Недостатки

- ❑ Единственный запрос не эффективен → последовательность детализированных запросов
- ❑ Достоверность ответов:
 - например, запрос цитаты подтверждающей какой-то факт → несуществующий автор, несуществующая книга, цитата относится к другой теме.
- ❑ Вероятностная модель: ответ зависит от контекста
- ❑ Баланс Популярность-Специфичность



ChatGPT. Возможности

- ❑ Поиск: Можно быстро находить нужную информацию, качество зависит от темы
- ❑ Разделение запроса на части и формулирование новых запросов на основе ответов повышает эффективность.
- ❑ Объяснение сложного концепта на разных уровнях сложности, например, идея книги.
- ❑ Освоение новых навыков, например изучение иностранного языка (написание текстов на заданную тему).
- ❑ Быстрое изучение новых областей знаний: получение основных концепций, идей, взаимосвязей (быстрее, чем через интернет поиск).

Литература

- ❑ François Chollet. Deep Learning with Python, Second Edition, 2021
- ❑ Transformer в картинках (<https://habr.com/ru/post/486358/>)