education

Natural Language Processing

Нейронные сети в ML

Козулин Илья

Цель занятия:

- Понять как может нейросеть оперировать текстовыми данными
- Разобрать как исторически развивались методы получения векторных представлений слов
- Увидеть что из себя представляют и как обучаются трансформеры
- Посмотреть какие задачи могут решаться при помощи текстовых данных и в чем их особенности

Регламент на занятия 29.10 и 05.11

- Ставим в чат
  - “++” если материал предельно понятен/очевиден/уже известен
  - “+” если материал более менее понятен, либо нужно еще немного над ним подумать, но можно идти дальше
  - “-” если что-то совсем не понятно и нужно остановиться и разобрать подробнее
- На лекции скидывается опрос с вопросами по материалу. Оценка за опрос идет к оценке за домашнее задание
- На семинаре будут практические задания, решаемые студентами с демонстрацией экрана. За каждое задание +1 балл к оценке за дз. Максимум +2 балла на одного человека.

О чём поговорим?

- Какие задачи охватывает NLP
- Векторные представления слов/предложений
- Моделирование языка
- Масштабирование моделей
- Parameter-Efficient Fine-Tuning

О чём поговорим?

- Какие задачи охватывает NLP
- Векторные представления слов/предложений

Часть 1

- Моделирование языка
- Масштабирование моделей
- Parameter-Efficient Fine-Tuning

Часть 2

О чём поговорим?

- Какие задачи охватывает NLP
- Векторные представления слов/предложений
- Моделирование языка
- Масштабирование моделей
- Parameter-Efficient Fine-Tuning

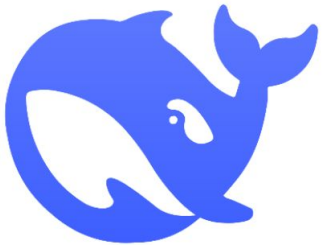
Часть 1

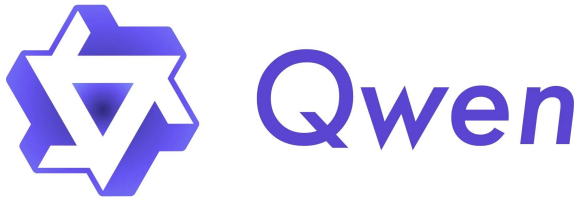
- Как все начиналось
- Где мы сейчас

Часть 2

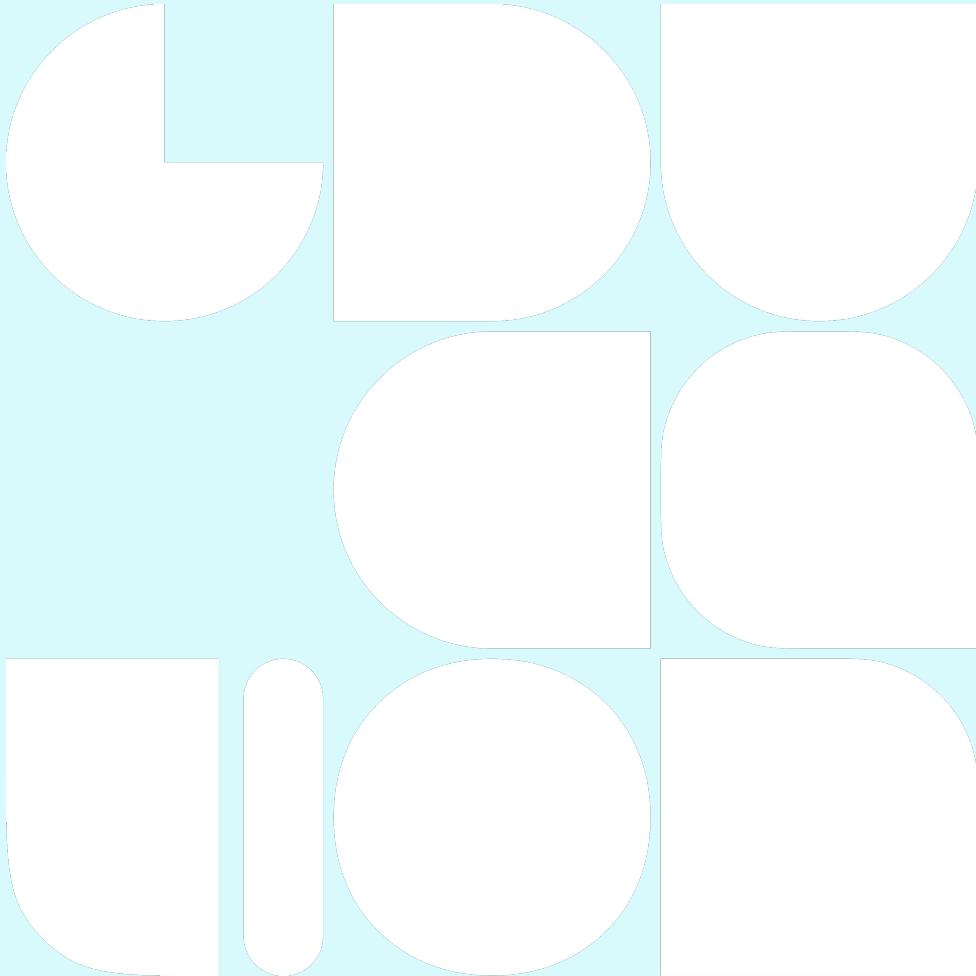
- Куда мы двигаемся

Введение





Какие задачи охватывает NLP



Классификация

1. Текстов/предложений

Примеры:

- I guess your parents dropped you a lot during childhood → “Hate Speech”
- It seems like the weather is going to be very unpleasant today → “Safe”

Классификация

1. Текстов/предложений

Примеры:

- I guess your parents dropped you a lot during childhood → “Hate Speech”
- It seems like the weather is going to be very unpleasant today → “Safe”

2. Токенов

Примеры:

- Named Entity Recognition (NER)

  Jürgen Schmidhuber published adversarial neural networks

  Jürgen | *PERSON (First Name)* Schmidhuber | *PERSON (Last Name)* published | *Common Word* adversarial | *Common Word* neural | *Common Word* networks| *Common Word*

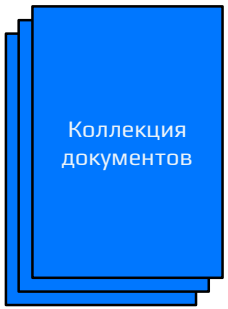
- Part-of-Speech (POS) Tagging

  Jürgen Schmidhuber published adversarial neural networks

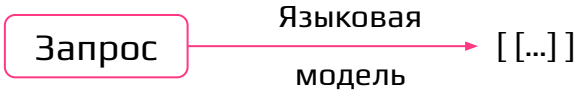
  Jürgen | *Proper Noun* Schmidhuber | *Proper Noun* published | *Verb* adversarial | *Adjective* neural | *Adjective* networks| *Common Noun*

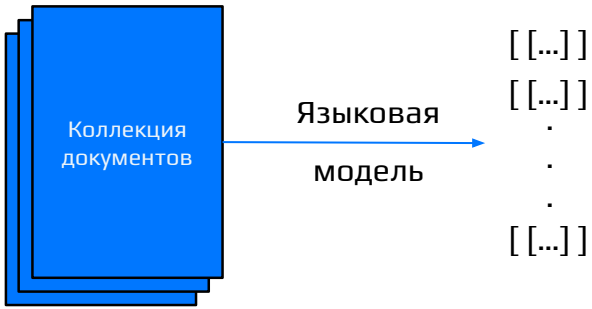
Семантический поиск (Semantic Search)

Запрос

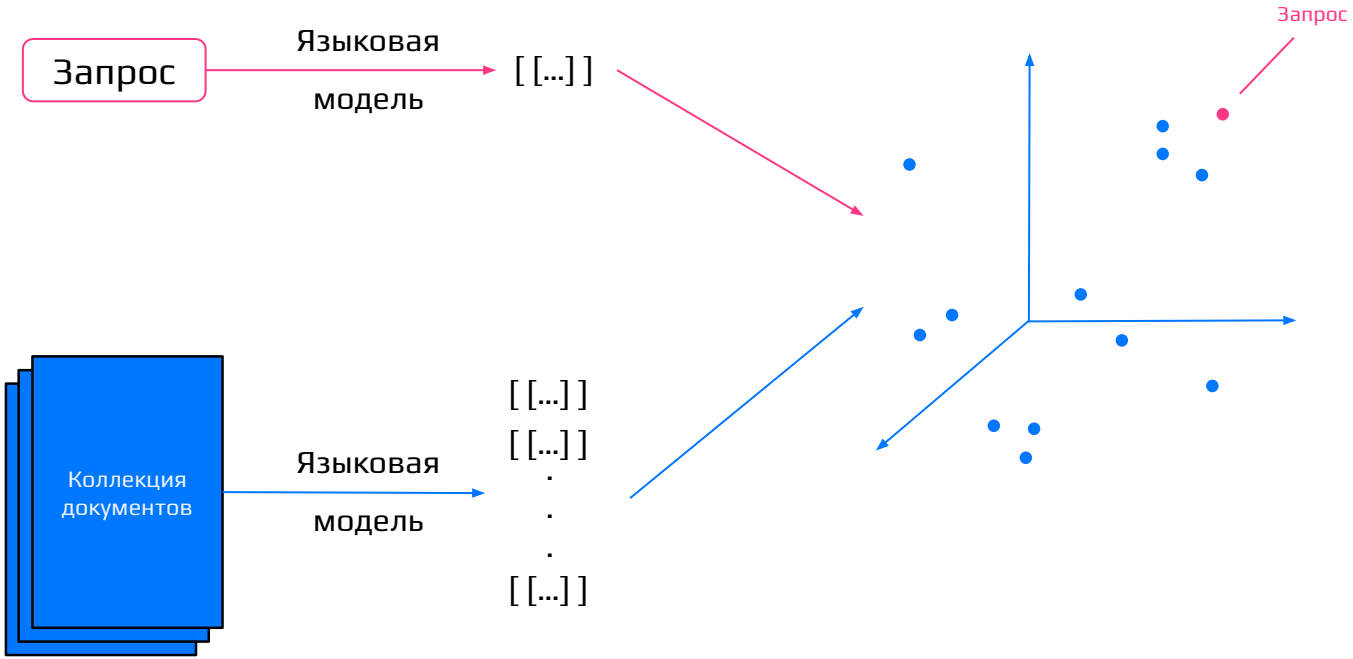


Семантический поиск (Semantic Search)

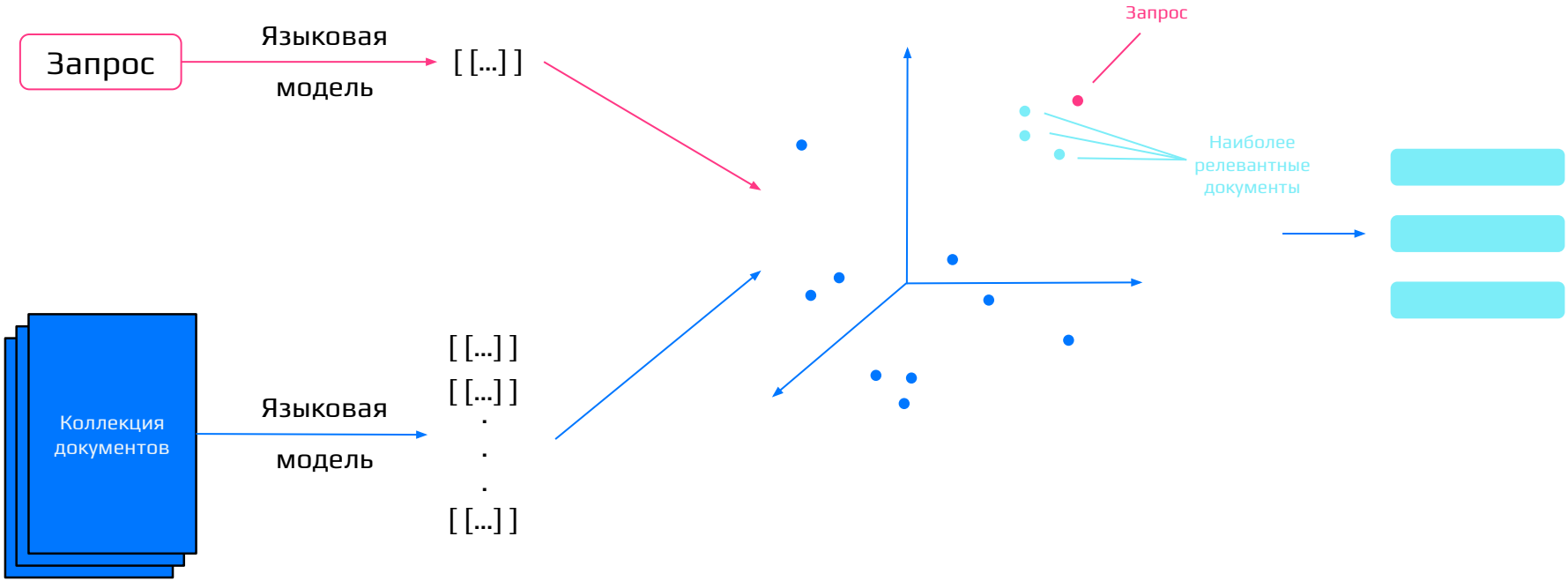




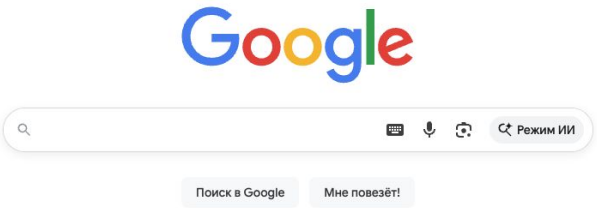
Семантический поиск (Semantic Search)



Семантический поиск (Semantic Search)



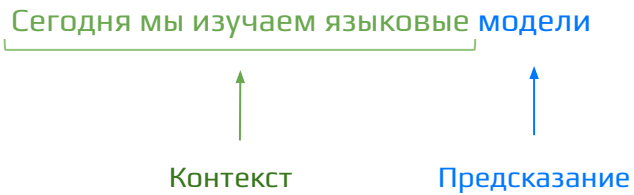
Семантический поиск (Semantic Search)



RAG

LangChain

Моделирование языка (Language Modeling)



Моделирование языка (Language Modeling)

Я к вам MASK — чего же боле? Что я могу еще сказать

Masked Language Modeling (MLM)

пишу

Модель видит все токены, включая те, что после маски

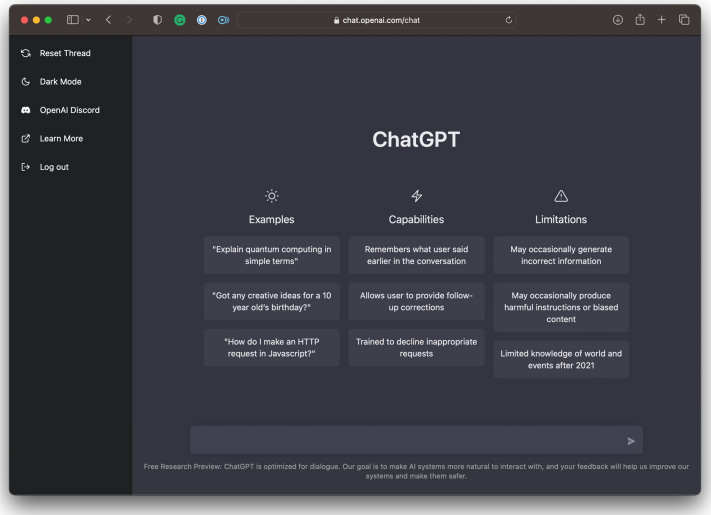
Я к вам пишу — чего же боле? Что я могу еще MASK

Causal Language Modeling (CLM)

сказать

Модель видит только те токены, что находятся до маски

Моделирование языка (Language Modeling)



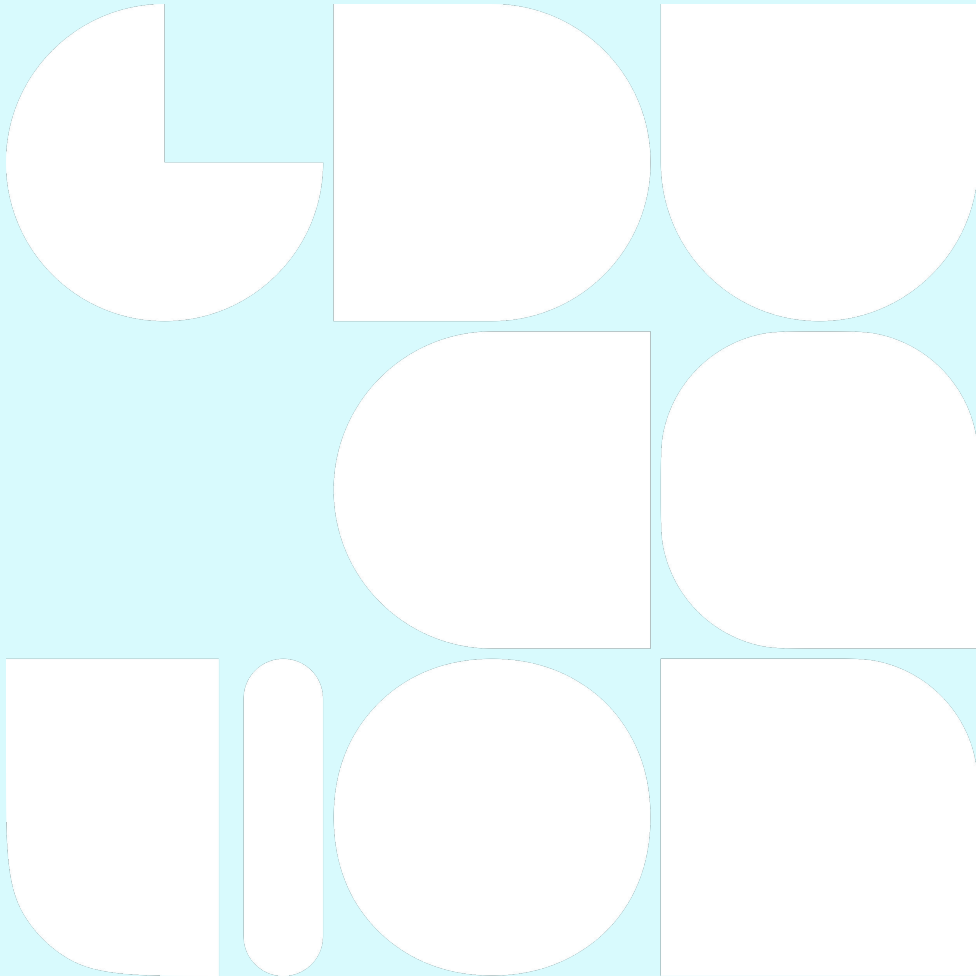
Итоги раздела:

- Задачи NLP очень разнообразны и тесно связаны с соседними доменами
- Обработка естественного языка сейчас крайне востребована и используется во многих известных продуктах
- Тексты можно обрабатывать с разной гранулярностью (полный текст/предложения/слова/токены/etc).

education

Вопросы?

Векторные представления слов/предложений



“Неглубокие” методы получения эмбеддингов слов

One-Hot Vectors

1) В прошлом году кот сбегал три раза
2) Быть или не быть, вот в чем вопрос

в	прошлом	году	кот	сбегал	три	раза	быть	или	не	вот	в	чем	вопрос
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1

“Неглубокие” методы получения эмбеддингов слов

One-Hot Vectors

Плюсы:

- Очень просто реализовать

Минусы:

- Всегда приходится хранить столько ключей, сколько всего уникальных слов во всех документах коллекции → огромная размерность
- Не учитывается контекст

“Неглубокие” методы получения эмбеддингов слов

Bag-of-Words

1) Мой кот не любит купаться. В прошлом году кот сбегал три раза

2) Быть или не быть, вот в чем вопрос

1)

мой	кот	не	любит	купаться	в	прошлом	году	сбегал	три	раза
1	2	1	1	1	1	1	1	1	1	1

2)

быть	или	не	вот	в	чем	вопрос
2	1	1	1	1	1	1

“Неглубокие” методы получения эмбеддингов слов

Bag-of-Words

Плюсы:

- Очень просто реализовать

Минусы:

- Всегда сильный перекос в сторону частиц и союзов (и, в, бы, но и т.д.)
- Не учитывается порядок слов
- В худшем случае приходится только на один документ хранить столько ключей, сколько слов в нем

“Неглубокие” методы получения эмбеддингов слов

TF-IDF

Идея: Если слово часто встречается во всех документах, то вряд ли эти слова имеют большое значение. И наоборот, если слово встречается редко, то скорее всего оно в большей степени определяет смысл.

$$\mathrm{tf}(t,d) = \frac{n_t}{\sum_k n_k}$$

Term Frequency

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{\, d_i \in D \mid t \in d_i \,\}|}$$

Inverse Document Frequency

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

n_t - число вхождений слова t в документ
D - коллекция документов

“Неглубокие” методы получения эмбеддингов слов

TF-IDF

Плюсы:

- Учитывает частоту слов как внутри документа, так и относительно всей коллекции
- Просто реализуется
- Можно интерпретировать как “веса” слов/токенов

Минусы:

- Все еще плохо хранит информацию о контексте

Word2Vec

Идея

Будем обучать векторные представления слов так, чтобы по ним можно было определить контекст

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Word2Vec

Идея

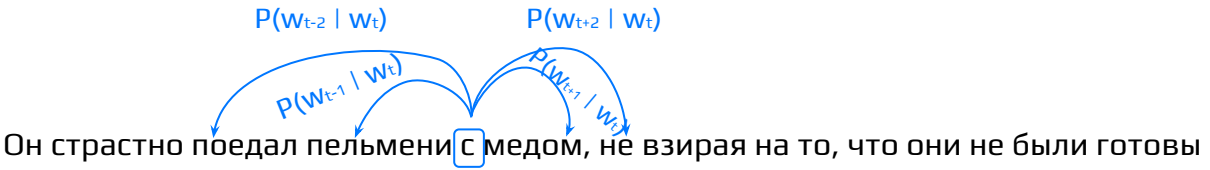
Будем обучать векторные представления слов так, чтобы по ним можно было определить контекст

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Word2Vec

Идея

Будем обучать векторные представления слов так, чтобы по ним можно было определить контекст



Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Word2Vec

Идея

Будем обучать векторные представления слов так, чтобы по ним можно было определить контекст

Он страстно поедал пельмени с медом, не взирая на то, что они не были готовы

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Word2Vec

Objective

Likelihood = $$L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \le j \le m, \\ j \neq 0}} P(w_{t+j} | w_t, \theta)$$

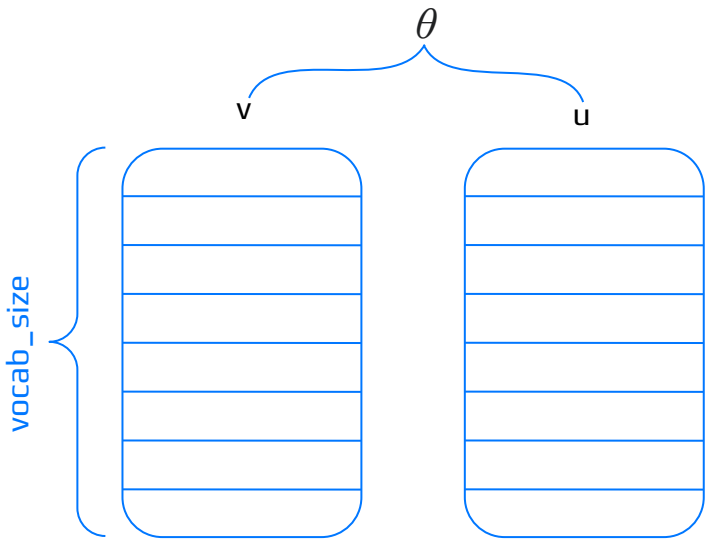
Loss = $$J(\theta) = -\frac{1}{T} \log L(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \sum_{\substack{-m \le j \le m, \\ j \neq 0}} \log P(w_{t+j} | w_t, \theta)$$

Word2Vec

Objective

Likelihood = $L(\theta) = \prod_{t=1}^{T} \prod_{-m \le j \le m, j \neq 0} P(w_{t+j}|w_t, \theta)$

Loss = $J(\theta) = -\frac{1}{T}\log L(\theta) = -\frac{1}{T}\sum_{t=1}^{T} \sum_{-m \le j \le m, j \neq 0} \log P(w_{t+j}|w_t, \theta)$



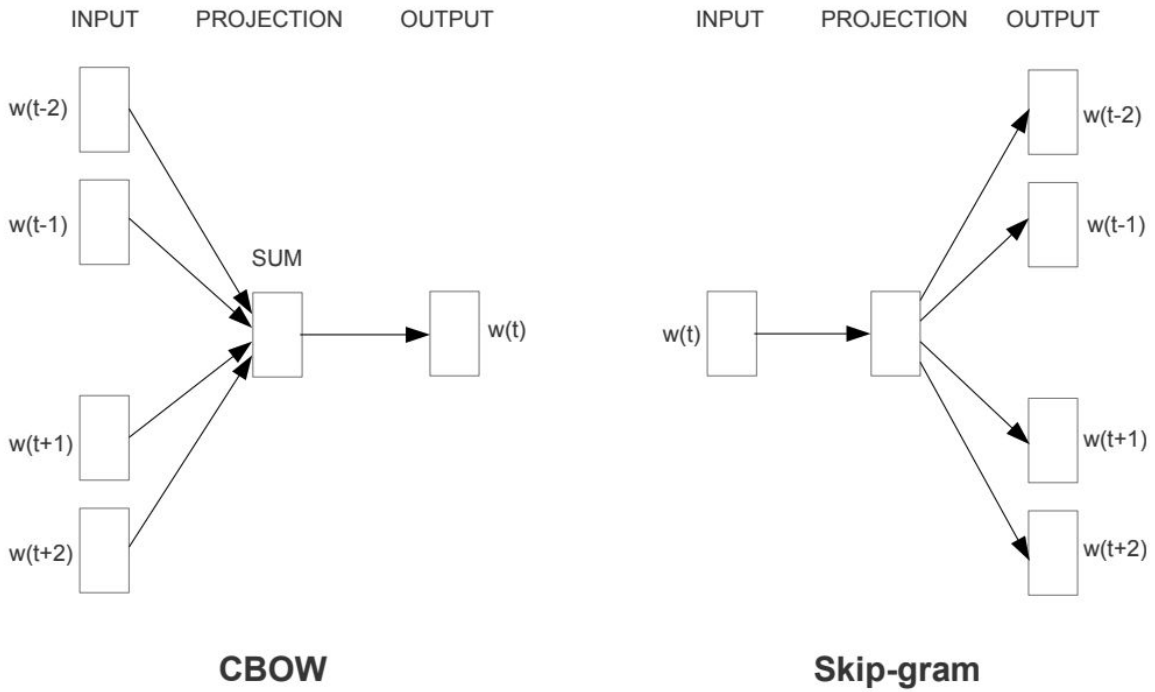
$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

Векторы центральных слов

Векторы центральных контекстных слов

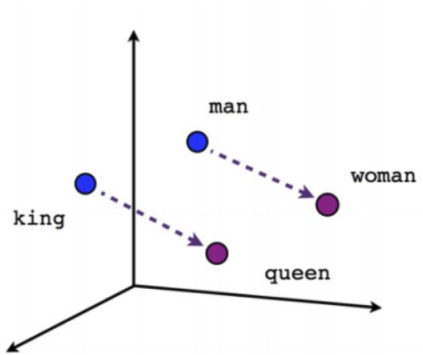
Word2Vec

Continuous Bag-of-Words (CBOW) vs Skip-Gram

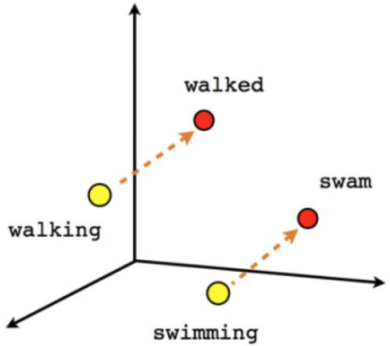


Word2Vec

Арифметика эмбеддингов



Male-Female



Verb tense

$$e(\text{«king»}) - e(\text{«man»}) + e(\text{«woman»}) \simeq e(\text{«queen»})$$
$$e(\text{«swimming»}) + e(\text{«walked»}) - e(\text{«walking»}) \simeq e(\text{«swam»})$$

Word2Vec

Плюсы:

- Умеет хранить семантическую информацию о словах

Минусы:

- Обычно использует целые слова, что сильно подгоняет модель под заданный корпус текстов. Чтобы использовать модель на новом наборе данных, нужно ее снова обучать на нем
- Все еще не использует всю информацию о контексте. Только в пределах заданного скользящего окна.

fasttext

Идея: используем в Word2Vec вместо слов n-граммы

Пример: 3-граммы

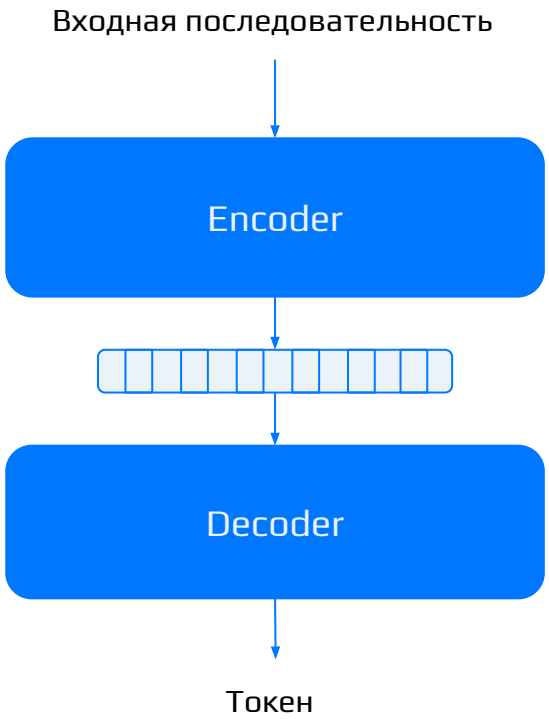
“Нейросеть” → [“Ней”, “ейр”, “йро”, “рос”, “осе”, “сет”, “еть”]

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135-146.

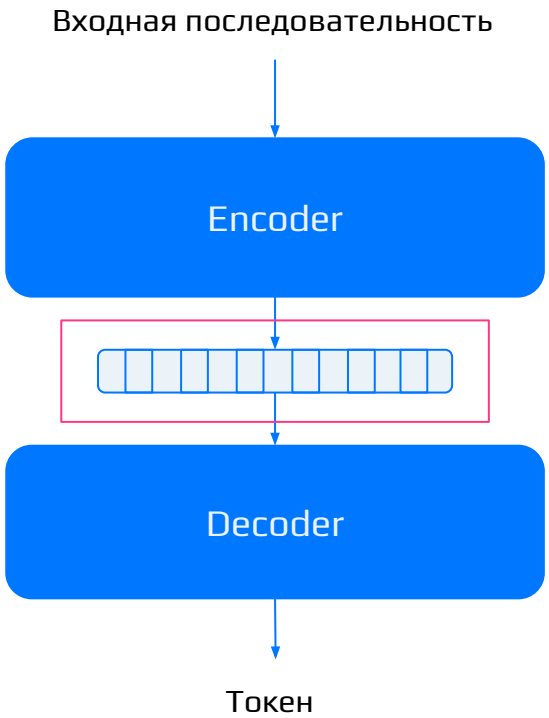
education

Вопросы?

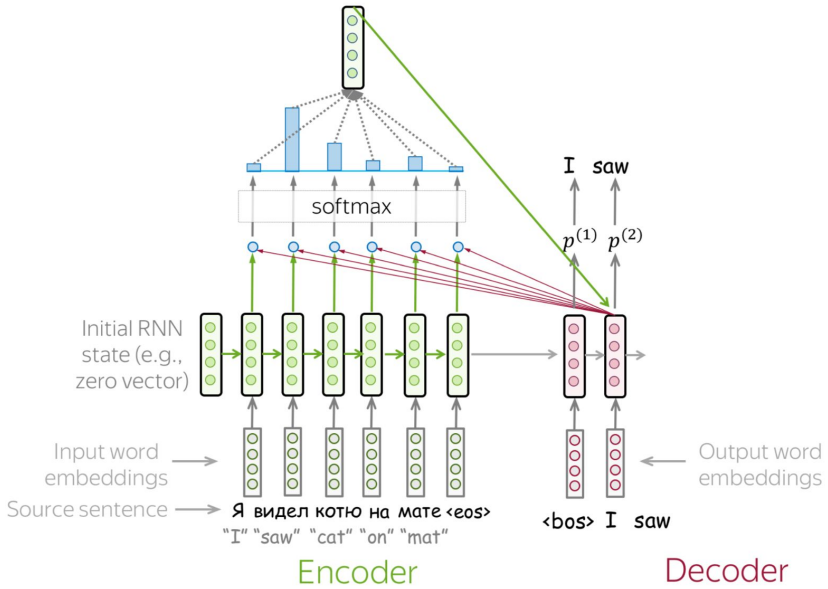
Attention



Attention

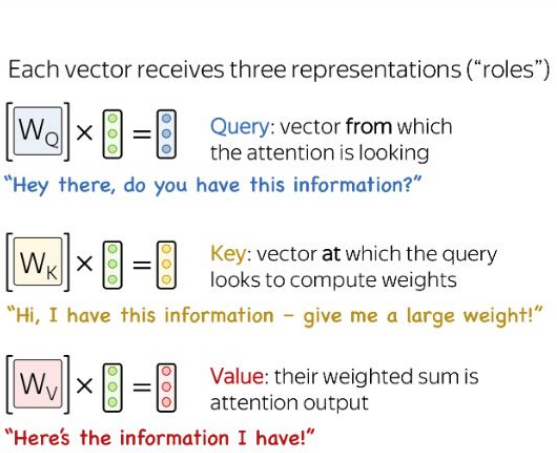


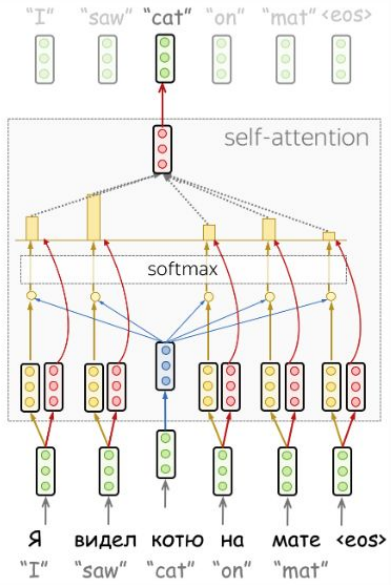
Attention



Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Self-Attention

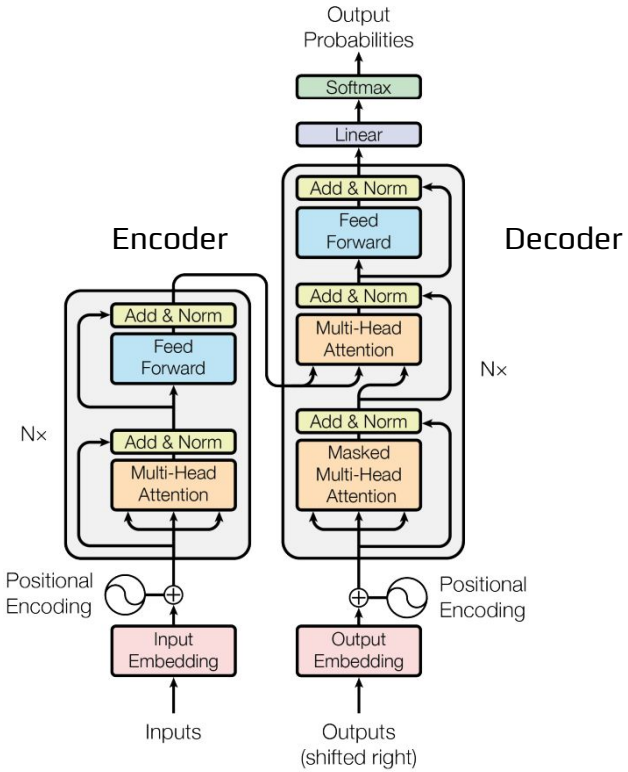




$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

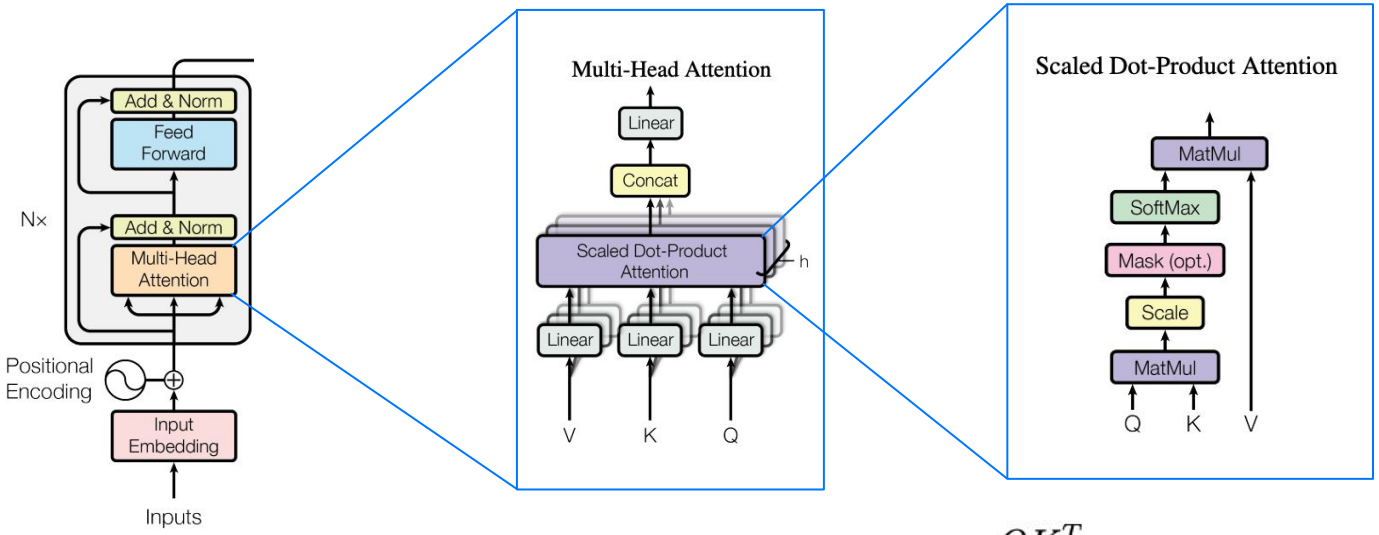
Layer Type	Complexity per Layer
Self-Attention	$O(n^2 \cdot d)$
Recurrent	$O(n \cdot d^2)$
Convolutional	$O(k \cdot n \cdot d^2)$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$

Transformer



Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Transformer

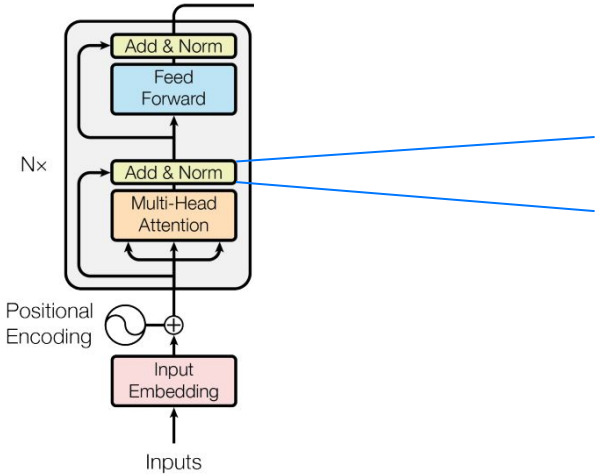


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

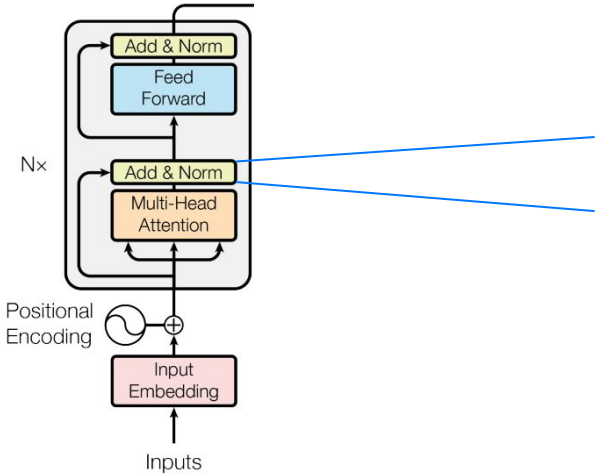
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Transformer

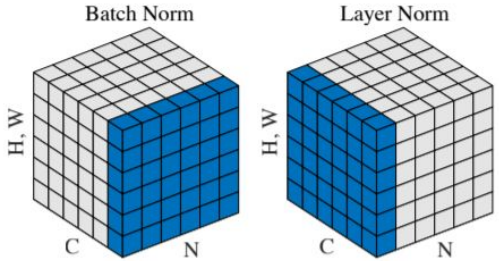


Add == Skip-connection element-wise addition
Norm == LayerNorm

Transformer

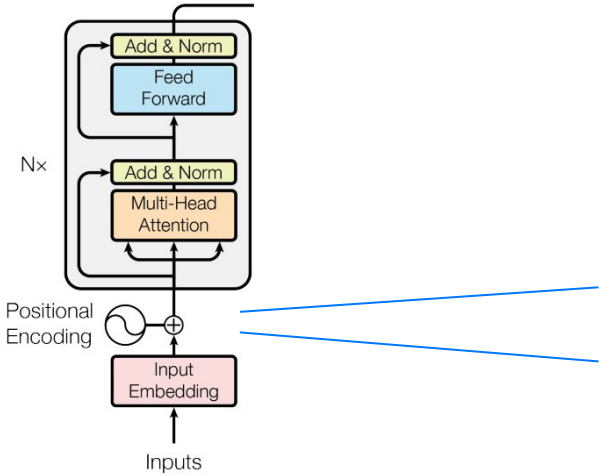


Add == Skip-connection element-wise addition
Norm == LayerNorm



$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

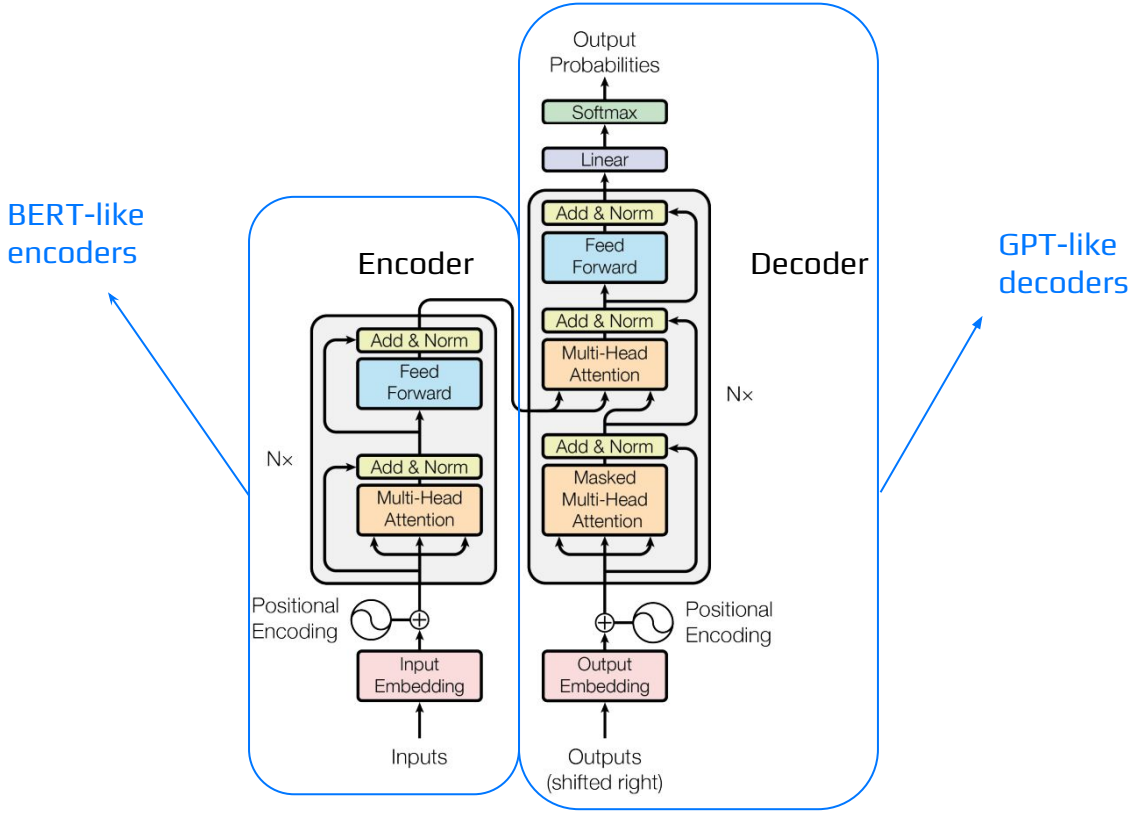
Transformer



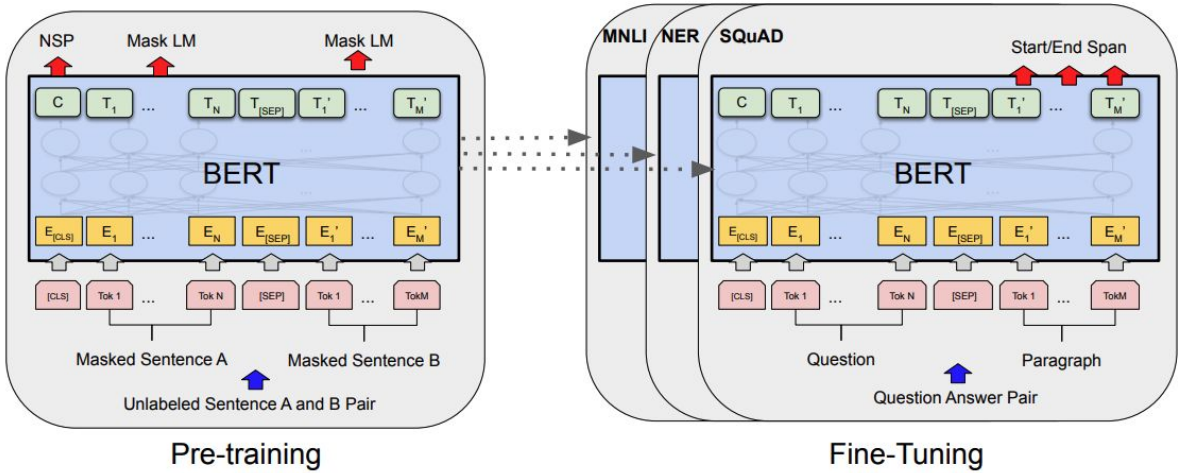
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$

$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

Transformer



Bidirectional Encoder Representations from Transformers (BERT)



- MASK - некоторые слова заменяем на токен неизвестного слова и пытаемся их восстановить.
- NSP - Для пары предложений пытаемся предсказать, правда ли, что B следует за A. (берем B случайно в 50% случаев)
- Нужно для улучшения модели языка и вопросно-ответных задач. Обучающее множество включает всю английскую википедию и книги не защищенные авторским правом. Для большой модели надо 4 дня на 16-и cloud TPU.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019, June). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)* (pp. 4171-4186).

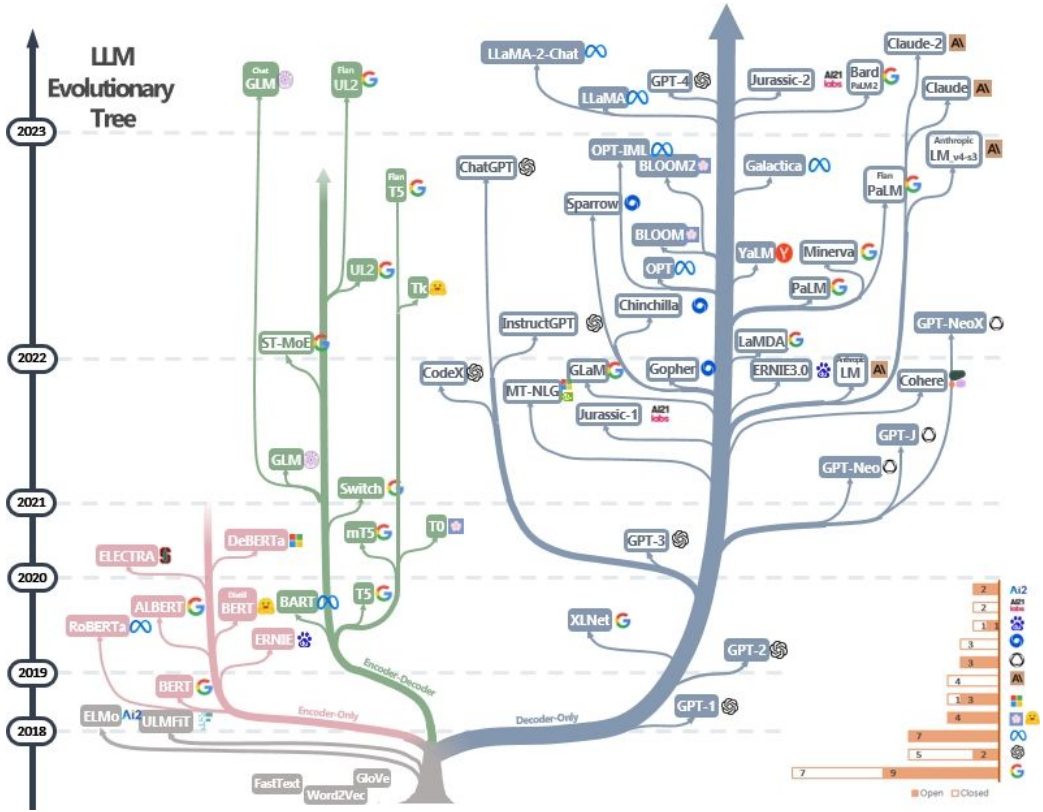
education

Вопросы?

Итоги раздела:

- Рассмотрены все основные неглубокие подходы к формированию признаков на основе текстов
- Пройден путь от Word2Vec до BERT
- Разобраны attention, self-attention и все основные компоненты трансформеров

Что дальше?


LLM
Evolutionary
Tree
2023
2022
2021
2020
2019
2018
GLM
UL2
LLaMA-2-Chat
GPT-4
LLaMA
Jurassic-2
Bard
Claude-2
Claude
OPT-IML
BLOOMZ
ChatGPT
Galactica
LM v4-s3
Flan T5
Sparrow
PaLM
BLOOM
YaLM
Minerva
OPT
Tk
PaLM
Chinchilla
InstructGPT
GPT-NeoX
LaMDA
ST-MoE
CodeX
GLaM
Gopher
ERNIE3.0
LM
Cohere
MT-NLG
Jurassic-1
GPT-J
GPT-Neo
GLM
Switch
mT5
T0
GPT-3
DeBERTa
ELECTRA
ALBERT
BERT
BART
T5
RoBERTa
ERNIE
XLNet
GPT-2
BERT
Encoder-Decoder
Encoder-Only
Decoder-Only
ELMo
ULMFiT
GPT-1
FastText
GloVe
Word2Vec
Open
Closed

Что дальше?

На следующей неделе

- Causal Language Modeling & GPT 1-3
- Метрики для оценки качества языковых моделей
- Становление LLM
- KV-cache, paged attention, sliding-window attention, prefix-caching
- PEFT (LoRA, p-tuning, prefix-tuning)

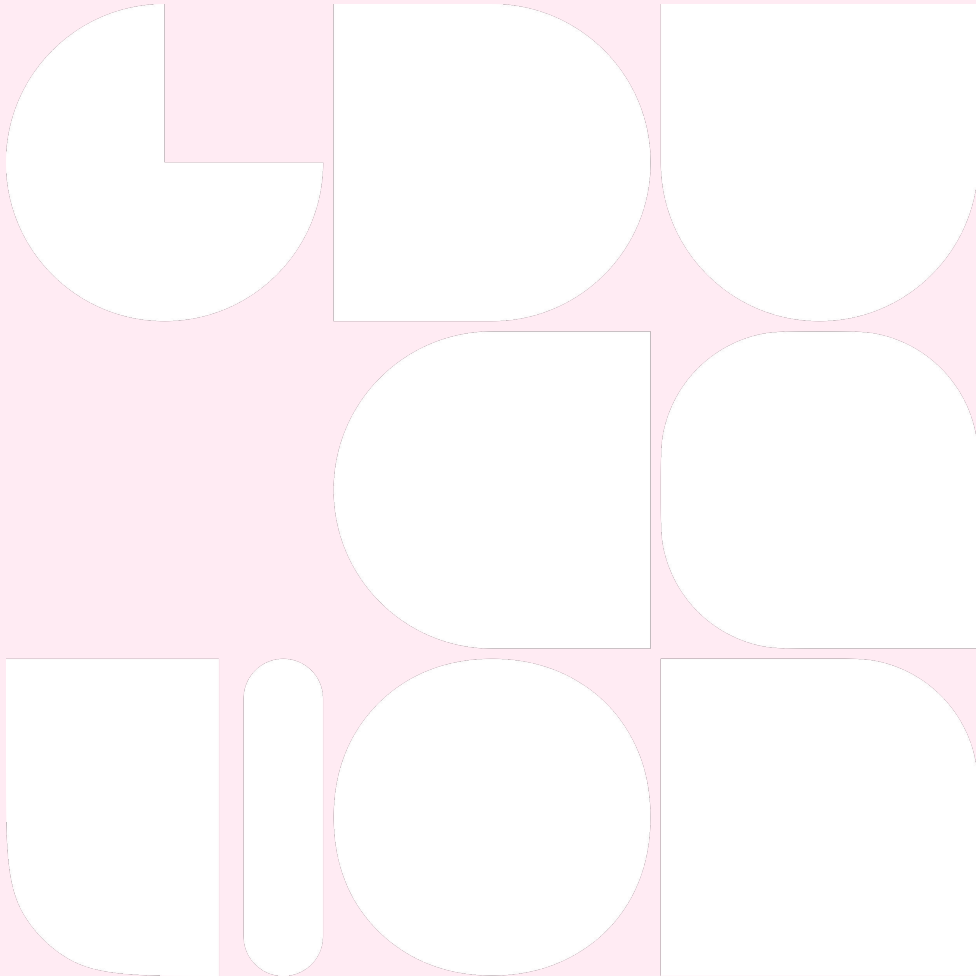
Подведём итоги:

- NLP покрывает все, что связано с векторными представлениями текстов, семантическим поиском и моделированием естественного языка
- Есть ряд необучаемых методов, которые до сих пор актуальны
- Word2Vec – первая нейросетевая модель для текстовых эмбеддингов, fasttext – его улучшенная версия на основе n-грамм
- Для учета информации из разных частей контекста используется оператор Attention
- Применяя Attention ко всем состояниям без RNN/CNN можно получить более гибкий учет контекста
- Трансформер – основная база всех современных языковых моделей, как для Masked LM, так и для Causal LM
- BERT – одна из ведущих архитектур среди моделей для получения текстовых эмбеддингов

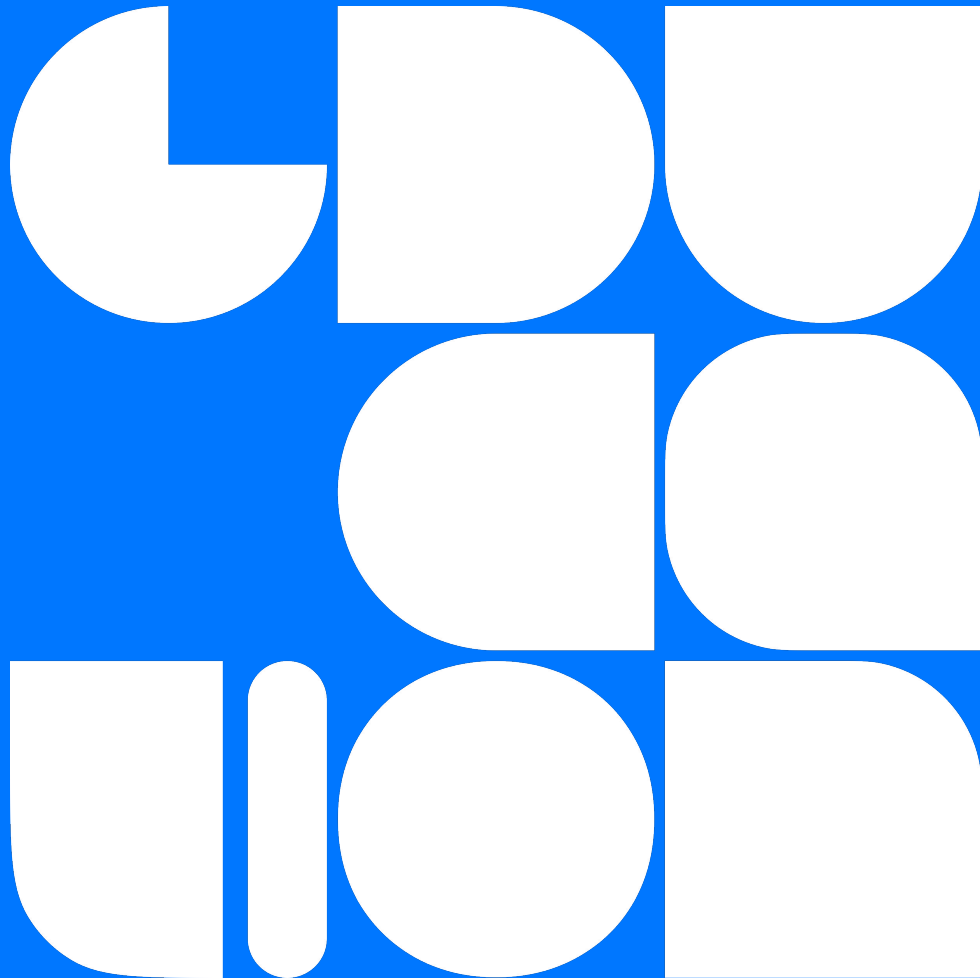
education

Вопросы?

Перерыв



Практика



Спасибо за внимание!

education