



Методы предобработки и векторизации текстов. Языковые модели.



Предобработка текста

- Для построения моделей машинного обучения необходимо предварительно предобработать текст. На этапе предобработки, как правило, используется только лингвистическая информация. Предобработка может включать следующие шаги:
 1. Уровень символов:
 - Токенизация – разбиение текста на токены (как правило, предложения или слова).
 2. Уровень слов:
 - Нормализация текста – лемматизация или стемминг.
 - Если текст получен путем генерации из аудио, то возможно к словам придется применять фонетические алгоритмы и искать наиболее похожие в словаре слова с использованием расстояния Левенштейна.
 - Частеречная разметка (Part-Of-Speech tagging, POS-tagging).
 - Выделение (распознавание) именованных сущностей, named-entity recognition (NER).
 3. Уровень предложений:
 - Разбор предложения, выделение семантических ролей.
- *Пример предобработки текста.*

Векторные представления на основе модели мешка слов

- Модель мешка слов.
- Использование CountVectorizer для построения мешка слов.
- Метрика TF-IDF.
- Использование TfidfVectorizer для построения мешка слов.
- *Пример решения задачи анализа тональности текста с помощью модели мешка слов.*

Неглубокие семантические векторные представления слов и документов

- Для рассматриваемых подходов используется термин «неглубокие векторные представления», так как в настоящее время существуют более сложные модели векторизации (например, BERT), основанные на глубоком обучении.
- [Обзор моделей](#)
- Семантические векторные представления слов:
 - Модель word2vec
 - Модель GloVe
 - Модель fastText
- Семантические векторные представления документов:
 - Модель doc2vec

Модель word2vec – описание

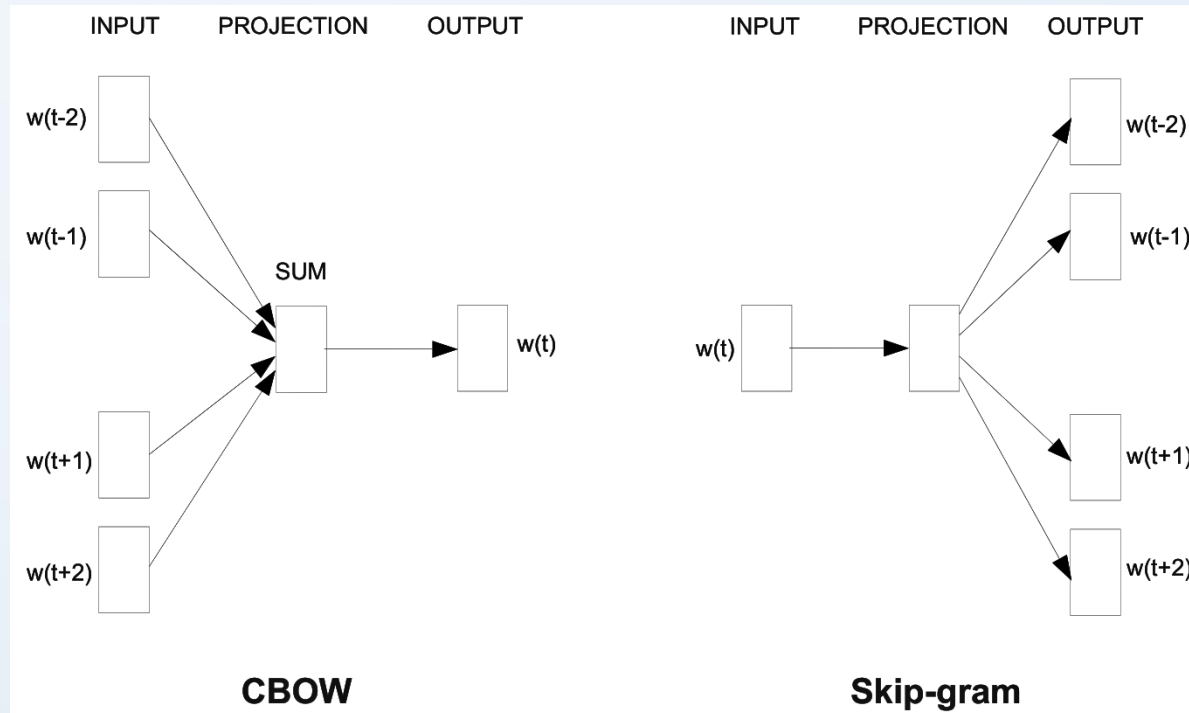
- Принцип работы:

- Модель word2vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе.
- Сначала генерируется словарь корпуса, а затем вычисляются векторные представления слов, «обучаясь» на входных текстах.
- Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие (по косинусному расстоянию) векторы. Полученные векторные представления слов могут быть использованы для обработки естественного языка и машинного обучения.
- Word2vec выполняет прогнозирование на основании контекстной близости слов. Так как инструмент word2vec основан на обучении нейронной сети, чтобы добиться его наиболее эффективной работы, необходимо использовать большие корпуса для обучения, что позволяет повысить качество предсказаний.

- Источники:

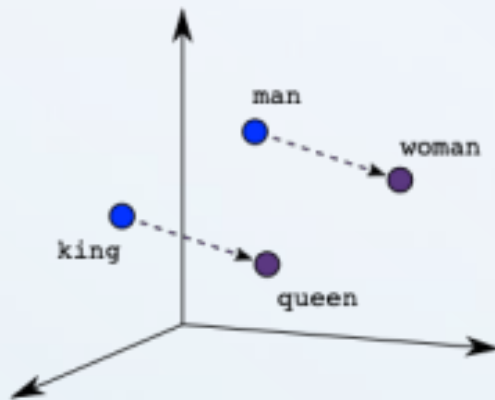
- Х. Лейн, Х. Хапке, К. Ховард. Обработка естественного языка в действии. — СПб.: Питер, 2020.
- [статья на русском языке](#),
- [оригинальная статья Т.Миколова](#),
- [пример визуализации](#).

Модель word2vec – алгоритмы обучения

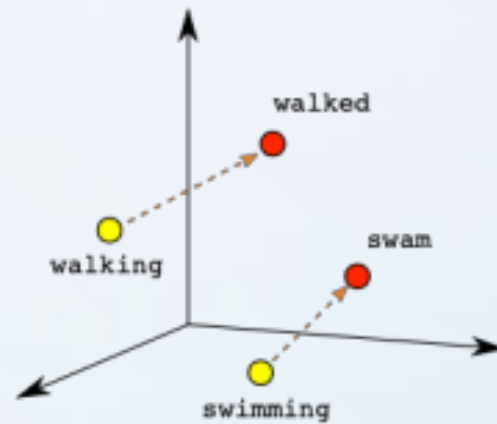


- Модель word2vec использует две возможные архитектуры:
 - CBoW — архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста.
 - Архитектура Skip-gram действует наоборот: она использует текущее слово, чтобы предсказывать окружающие его слова.
 - Построение модели word2vec возможно с помощью двух данных алгоритмов. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов.
 - Архитектура Skip-gram хорошо работает для маленьких корпусов и редко встречающихся термов. Архитектура CBoW демонстрирует более высокую точность для часто встречающихся слов, и обучение занимает гораздо меньше времени.

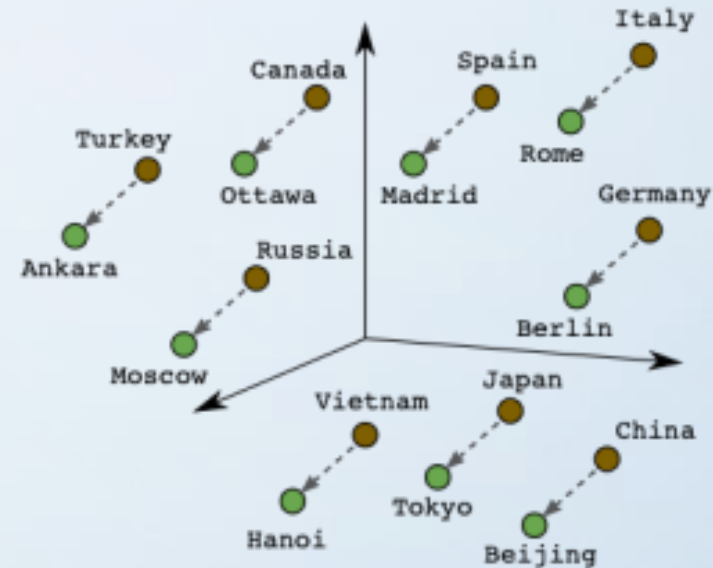
Модель word2vec – вычисления с векторами



Male-Female



Verb Tense



Country-Capital

Источник: <https://www.baeldung.com/cs/convert-word-to-vector>

- Получаемые на выходе векторные представления слов позволяют вычислять «семантическое расстояние» между словами. Так, можно находить похожие по значению слова.
- Пример: король относится к мужчине также, как королева к женщине.

Модель GloVe – описание

- Исследователи из Стэнфорда решили выяснить, почему Word2vec так хорошо работает, и найти оптимизируемую функцию стоимости. Они начали с подсчета совместных вхождений слов и занесения их в квадратную матрицу. Оказалось, что можно вычислить сингулярное разложение этой матрицы совместной встречаемости, разбив ее на те же две матрицы весов, что генерирует word2vec.
- Но в некоторых случаях модель Word2vec не сходится к тому глобальному минимуму, который получался у стэнфордских исследователей с помощью SVD. Именно от непосредственной оптимизации глобальных векторов (global vectors) совместной встречаемости слов (совместных вхождений в рамках всего корпуса) и получил свое название метод GloVe.
- Метод GloVe может формировать матрицы, эквивалентные входным и выходным матрицам весов Word2vec, в результате чего получается языковая модель с той же точностью, что Word2vec, но за намного меньшее время.
- Полученные представления отражают важные линейные подструктуры векторного пространства слов: получается связать вместе разные спутники одной планеты или почтовый код города с его названием.
- GloVe учитывает совместную встречаемость, а не полагается только на контекстную статистику. Векторы слов группируются вместе на основе их глобальной схожести.
- GloVe использует простую архитектуру без нейронной сети, поэтому векторные представления строятся быстрее, чем в случае word2vec. GloVe опережает word2vec на большинстве бенчмарков.
- Источники:
 - [оригинальная статья](#)

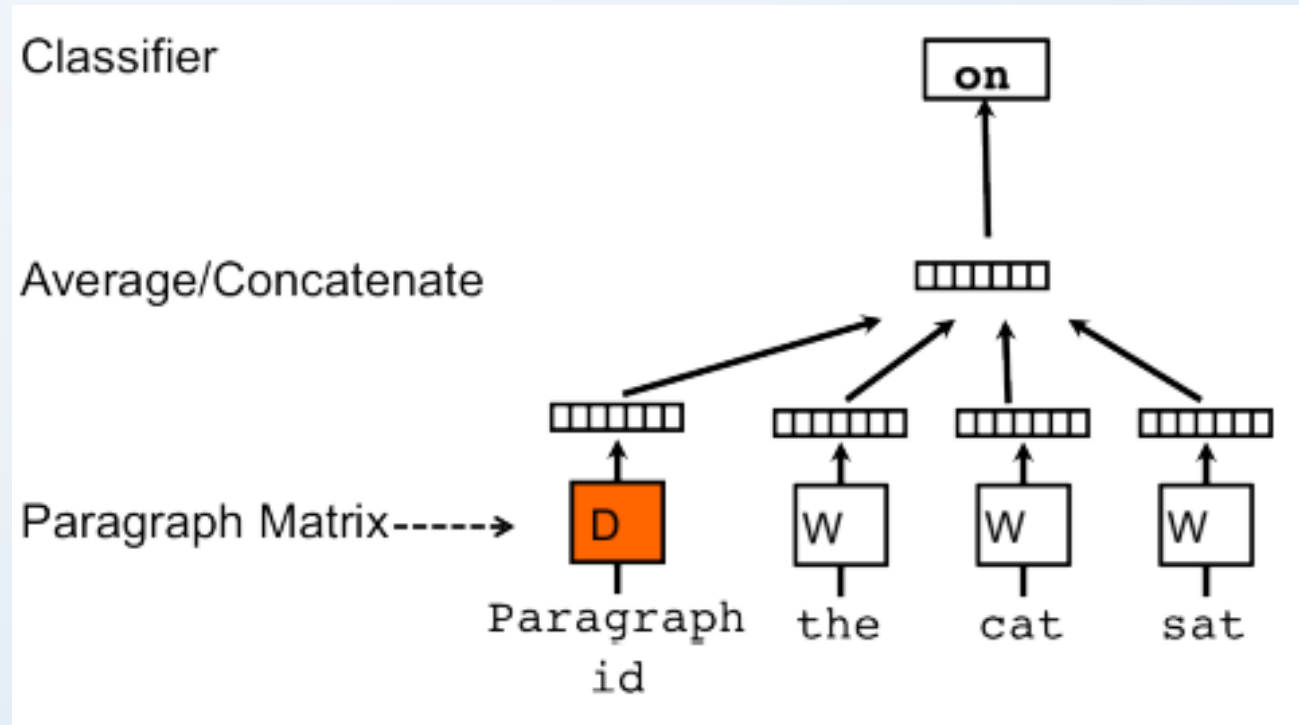
Модель fastText – описание

- Основной проблемой Word2Vec и GloVe является необходимость формирования фиксированного словаря слов перед началом работы модели. Появление новых слов, отсутствующих в словаре, является проблемой.
- Модель fastText – это расширение модели Word2Vec, предложенное авторами этой модели.
- К основной модели Word2Vec добавлена модель символьных n-грамм. Каждое слово представляется композицией нескольких последовательностей символов определённой длины. Например, слово they в зависимости от гиперпараметров может состоять из "th", "he", "ey", "the", "hey". Вектор слова – это сумма всех его n-грамм.
- Поскольку количество n-грамм может быть очень велико, то n-граммы хэшируются с использованием хэш-функции [Fowler-Noll-Vo](#).
- Далее могут использоваться подходы CBoW/Skip-gram для создания векторных представлений.
- Источники:
 - [Официальный сайт](#),
 - [оригинальная статья Т. Миколова](#),
 - Модели GloVe и fastText как улучшения Word2Vec – [статья](#),
 - FastText: рецепт работы по коду – [статья](#).
- *Пример использования моделей word2vec и fastText.*

Модель doc2vec – описание

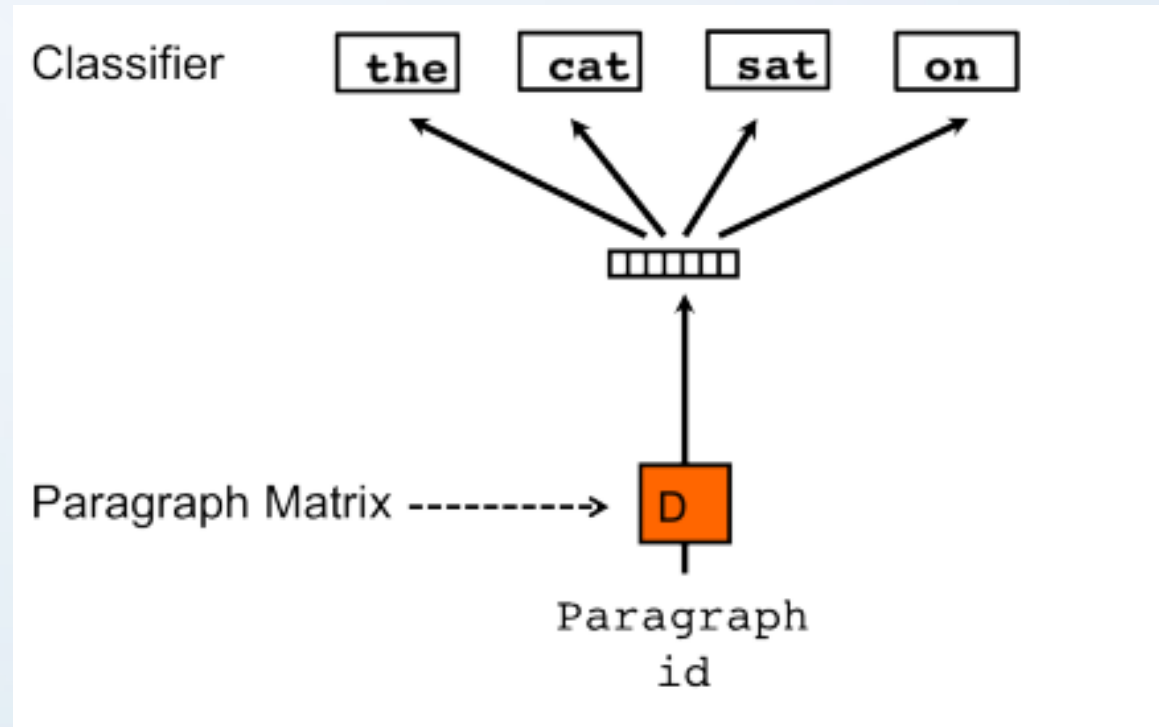
- По аналогии с векторными представлениями для слов, можно строить векторные представления для документов. В этом случае учитывается смысл не только отдельных слов, но и абзацев, предложений, документов.
- Модель представляет собой расширение модели Word2Vec, в которую дополнительно введен идентификатор параграфа - Paragraph Id (может использоваться также идентификатор документа или фрагмента документа).
- За счет расширения идеи Word2vec с помощью дополнительного вектора документа или параграфа, применяемого для предсказания слов, можно использовать полученный в результате обучения вектор документа для различных целей, например для поиска в корпусе схожих документов.
- Источники:
 - [статья с описанием модели](#),
 - [оригинальная статья авторов](#).

Модель doc2vec – алгоритмы обучения – PV-DM



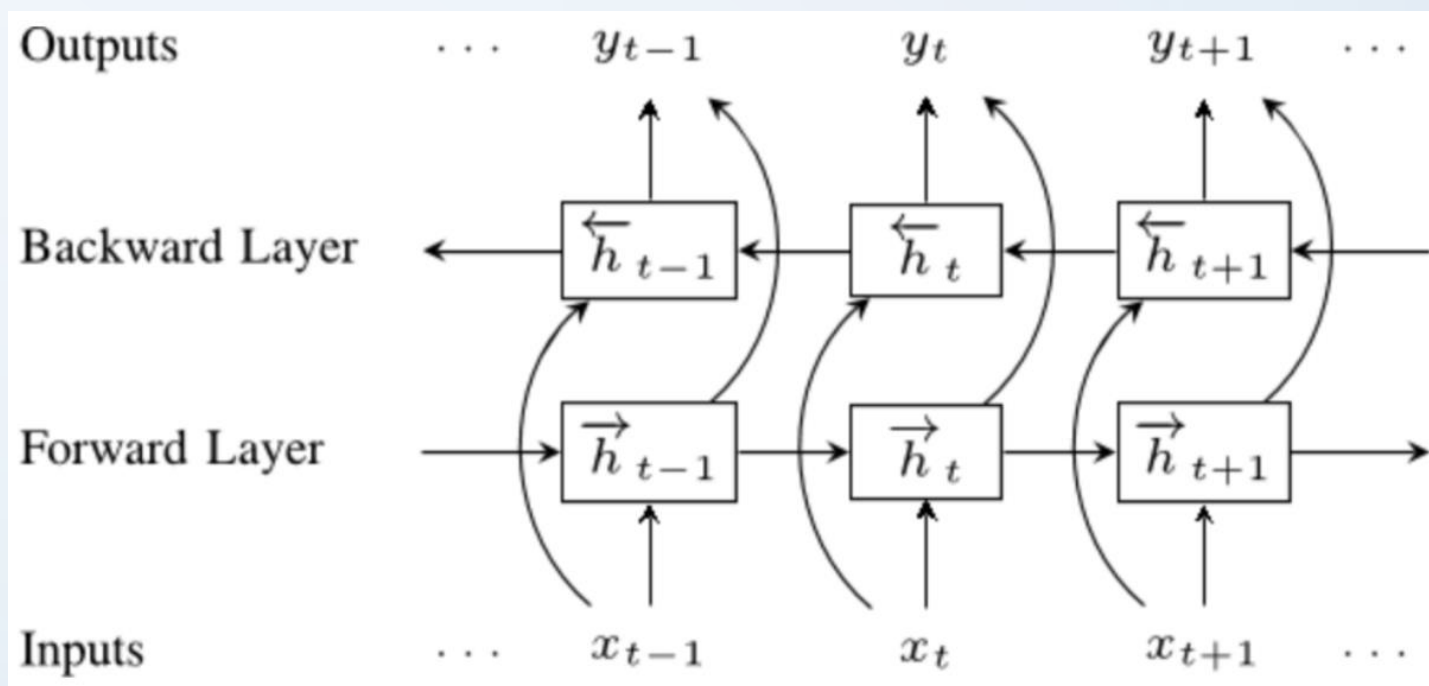
- В качестве расширения архитектуры CBoW используется архитектура PV-DM (Distributed Memory version of Paragraph Vector) — архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста с учетом идентификатора параграфа.

Модель doc2vec – алгоритмы обучения – PV-DBOW



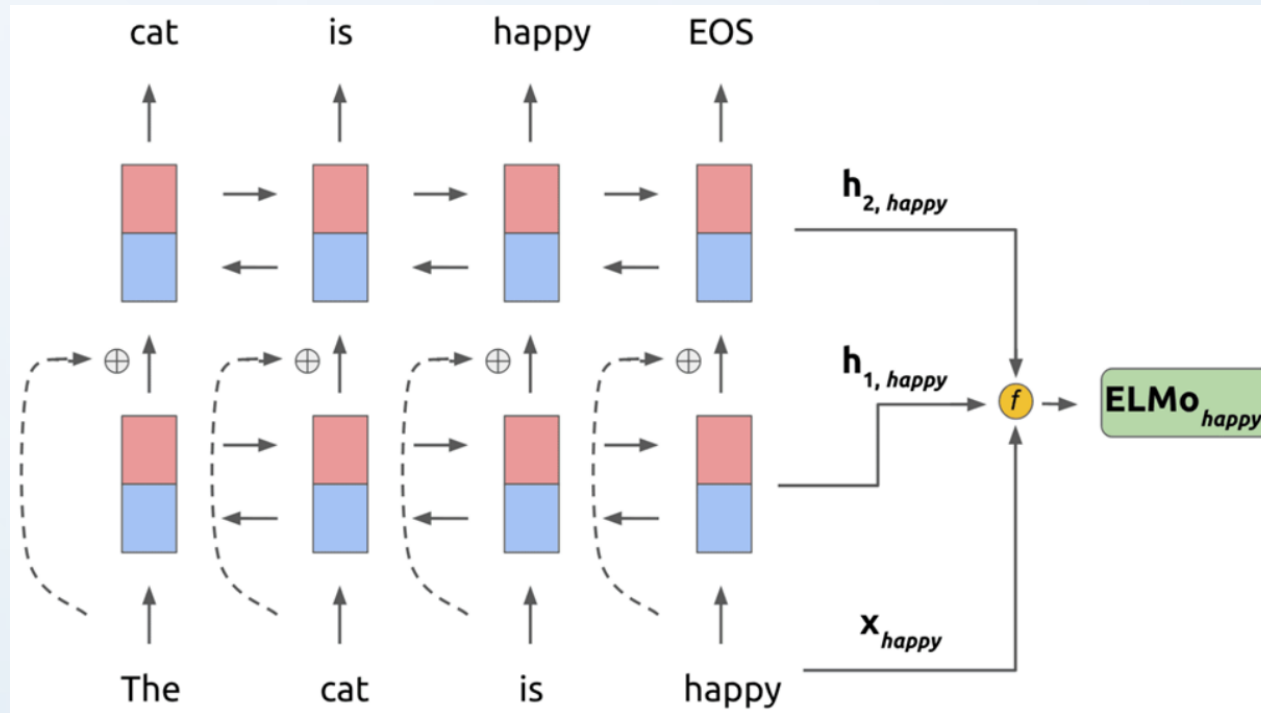
- Аналогом архитектуры Skip-gram является архитектура PV-DBOW (Distributed Bag of Words version of Paragraph Vector), которая предсказывает текст параграфа по его идентификатору.

Модель ELMo – описание и структура сети



- Строит векторные представления слов с учетом их встречаемости в контексте документа. Для этого используется двунаправленная LSTM-сеть (с прямым и обратным слоем), итоговый вектор слова является объединением векторов в каждой из сетей.
- Источники:
 - [оригинальная статья авторов](#),
 - [описание на paperswithcode](#),
 - [статья с пояснениями](#).

Модель ELMo – пример



- Пример формирования векторного представления слова показан на рисунке.
- Преимущества модели:
 - Векторные представления слов строятся с учетом как их синтаксиса, так и семантики.
 - Модель принимает во внимание контекст слова (в том числе с учетом полисемии).
- Примеры использования в Python: [первый](#), [второй](#), [третий](#).

Механизм внимания в нейронных сетях

- Механизм внимания используется для поиска взаимосвязей между различными частями ВХОДНЫХ и ВЫХОДНЫХ ДАННЫХ.
- В области NLP наиболее часто применяется механизм «самовнимания» (Self-Attention) – разновидность механизма внимания, задачей которой является выявление закономерности только между ВХОДНЫМИ ДАННЫМИ.
- Механизм Self-Attention позволяет учитывать связи между далекими словами в предложении и отказаться от использования рекуррентных нейронных сетей при обработке текстов.
- Источники:
 - Базовая статья «Neural Machine Translation by Jointly Learning to Align and Translate» - <https://arxiv.org/abs/1409.0473>
 - Базовая статья «Attention Is All You Need» - <https://arxiv.org/abs/1706.03762>
 - Русскоязычная статья - https://neerc.ifmo.ru/wiki/index.php?title=%D0%9C%D0%B5%D1%85%D0%B0%D0%BD%D0%B8%D0%B7%D0%BC_%D0%B2%D0%BD%D0%B8%D0%BC%D0%B0%D0%BD%D0%B8%D1%8F

Модель «Трансформер»

- Основана на архитектуре кодировщик-декодировщик, при этом и кодировщик и декодировщик используют механизм самовнимания.
- В базовой статье использовалась для решения задачи машинного перевода.
- Модель позволяет учитывать связи между далекими словами в предложении и отказаться от использования рекуррентных нейронных сетей при обработке текстов.
- Источники:
 - Базовая статья «Attention Is All You Need» - <https://arxiv.org/abs/1706.03762>
 - Блог компании Google - <https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html>
 - Русскоязычная статья - <https://habr.com/ru/post/341240/>

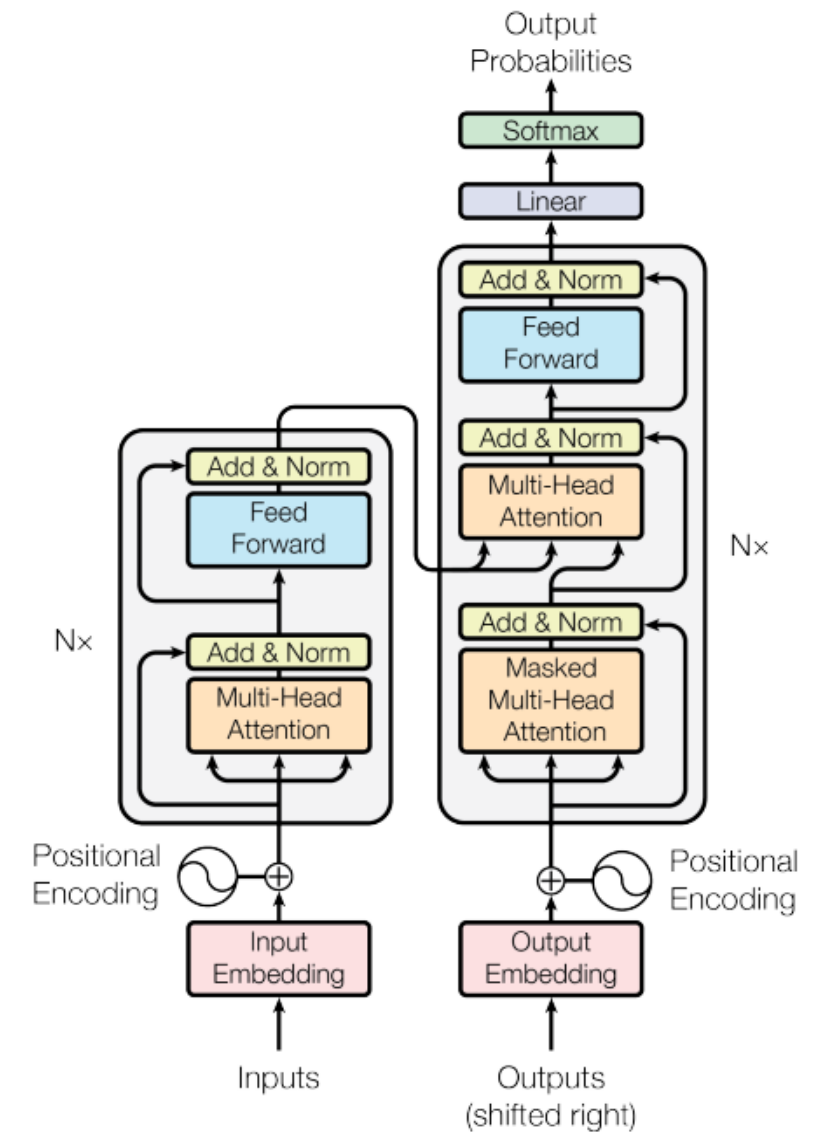
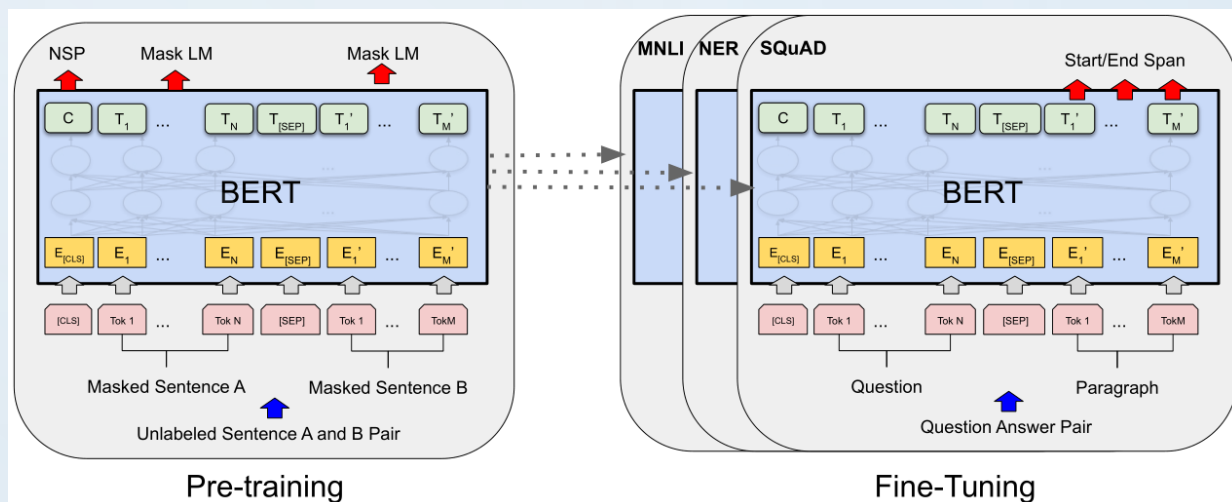


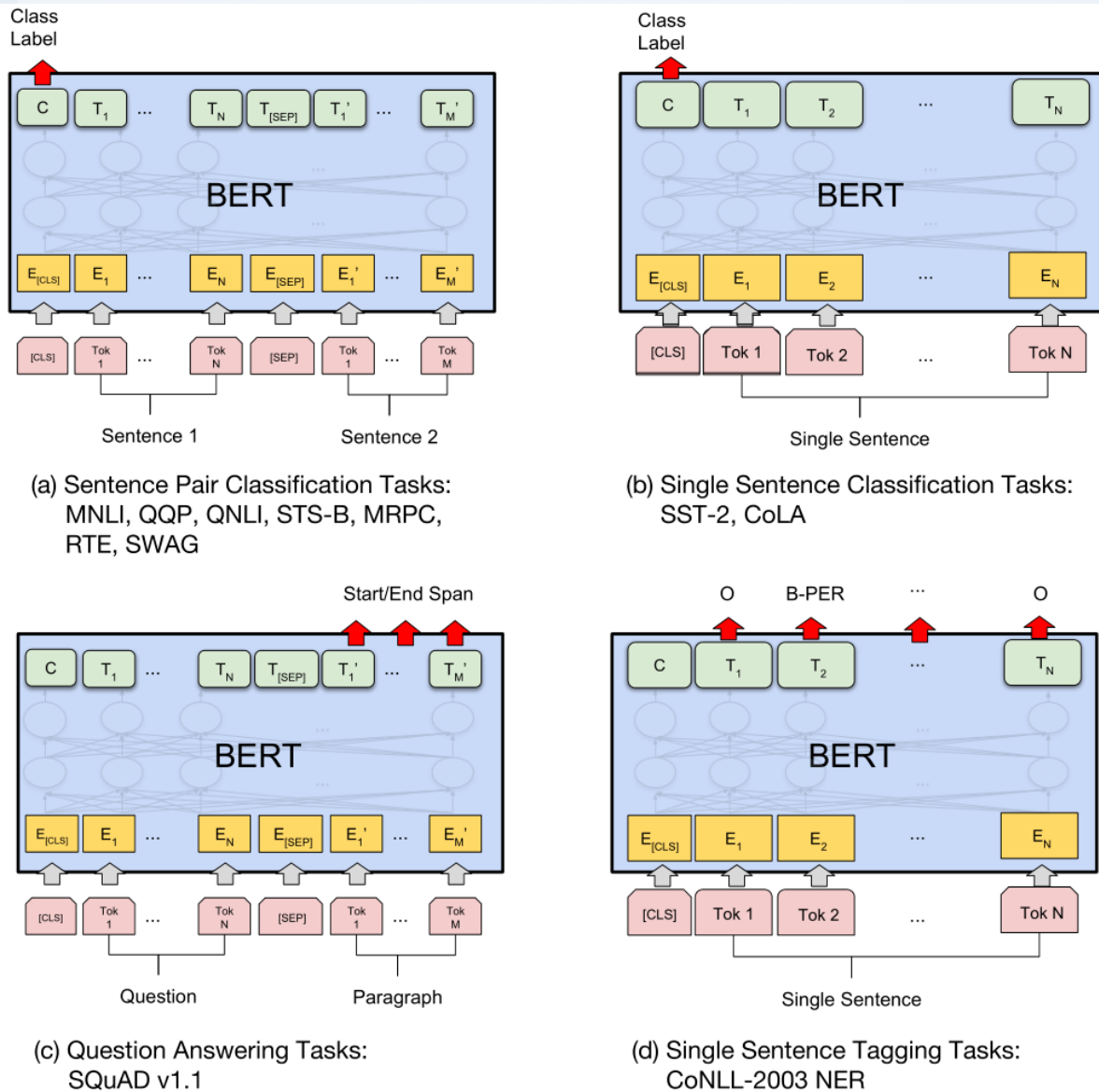
Figure 1: The Transformer - model architecture.

Модель BERT – описание

- Для модели BERT используется термин «языковая модель» (language model). Данная модель может применяться для решения нескольких NLP-задач.
- Основу BERT составляет композиция кодировщиков трансформера. BERT является автокодировщиком. В каждом слое кодировщика применяется двустороннее внимание, что позволяет модели учитывать контекст с обеих сторон от рассматриваемого токена, а значит, точнее определять значения токенов. Обучение модели включает два этапа:
 - Предобучение (pre-training). BERT обучается одновременно на двух задачах – предсказания следующего предложения (англ. next sentence prediction) и генерации пропущенного токена (англ. masked language modeling).
 - Точная настройка (fine-tuning). Выход сети, полученной на этапе предобучения, может использоваться как вход для решаемой задачи (например, задачи классификации текстов).
- Источники:
 - Базовая статья «BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding» - <https://arxiv.org/abs/1810.04805v2>
 - Блог компании Google - <https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html>
 - Статья «BERT Explained: State of the art language model for NLP» - <https://towardsdatascience.com/bert-explained-state-of-the-art-language-model-for-nlp-f8b21a9b6270>
 - Русскоязычная статья - [https://neerc.ifmo.ru/wiki/index.php?title=BERT_\(%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C\)](https://neerc.ifmo.ru/wiki/index.php?title=BERT_(%D1%8F%D0%B7%D1%8B%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C))
 - Русскоязычная статья с примерами использования - <https://habr.com/ru/post/436878/>



Модель BERT – варианты «точной настройки»

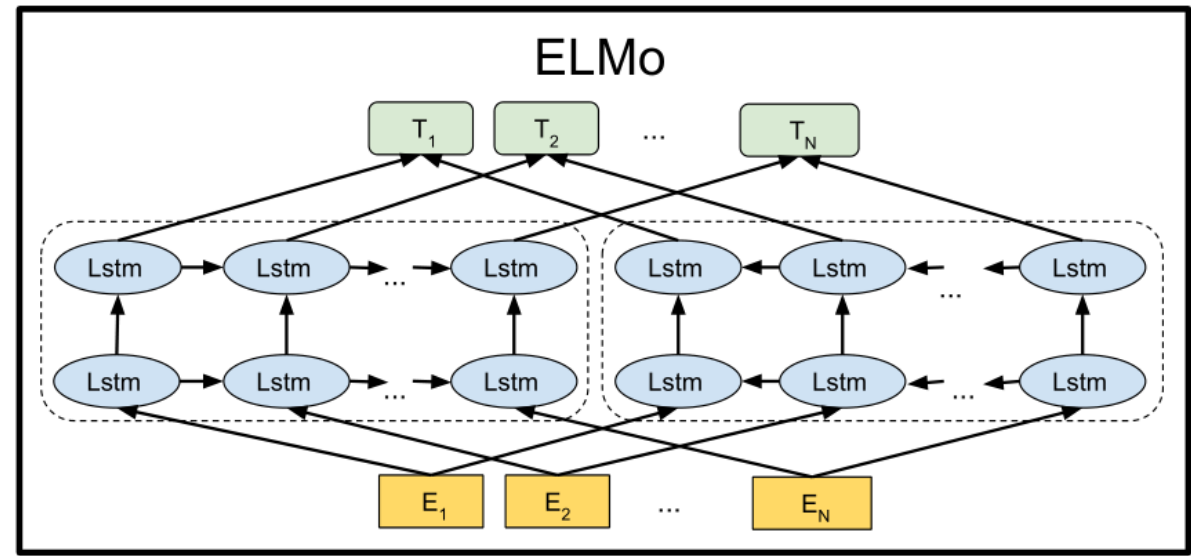
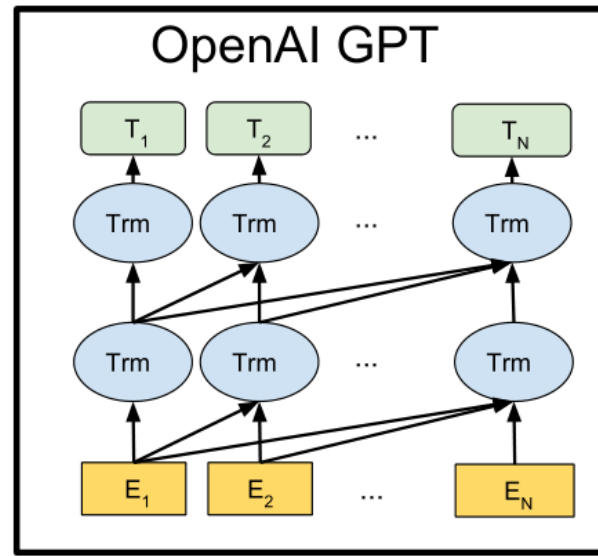
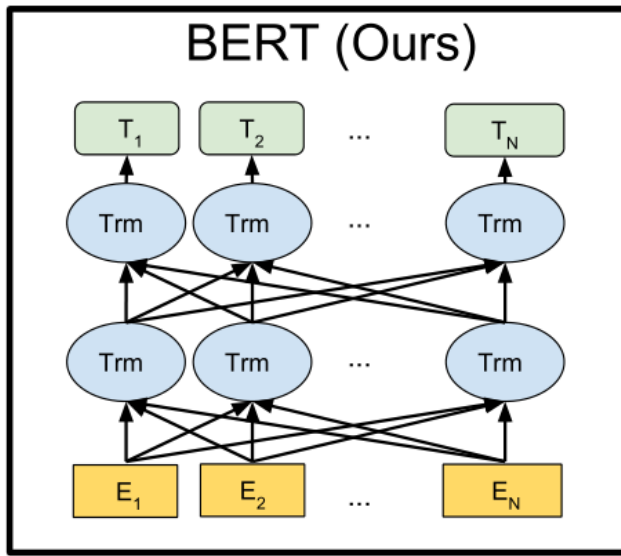


Решение задач:

- (a) – классификации пары предложений (например, является ли второе предложение следствием или опровержением первого).
- (b) – классификация одного предложения (например, корректно или некорректно оно составлено).
- (c) – выделение из второго предложения той части, которая является ответом на вопрос из первого предложения.
- (d) – разметка слов в предложении (например, выделение именованных сущностей).

Figure 4: Illustrations of Fine-tuning BERT on Different Tasks.

Модель BERT – сравнение с GPT и ELMo



- BERT использует двунаправленную трансформерную модель.
- OpenAI GPT использует трансформеры со связями слева-направо.
- ELMo использует конкатенацию векторов из двух LSTM-сетей, обученных слева-направо и справа-налево.

«Конкурент» модели BERT – XLNet

- Модель XLNet выдает на выходе вероятность совместной встречаемости последовательности токенов на основе двухпоточкового механизма самовнимания. Задачей обучения модели является подсчет вероятности появления заданного слова (токена), при условии наличия всех других слов (токенов) в предложении.
- В отличие от модели BERT, модель XLNet учитывает все возможные перестановки слов в предложении.
- Как правило, качество у модели XLNet лучше чем у BERT.
- Источники:
 - Базовая статья «XLNet: Generalized Autoregressive Pretraining for Language Understanding» - <https://arxiv.org/abs/1906.08237>
 - Статья «XLNet: новый state-of-the-art в задачах обработки естественного языка» – <https://neurohive.io/ru/papers/xlnet-novyj-state-of-the-art-v-zadachah-obrabotki-estestvennogo-yazyka/>
 - Статья «Разбираем XLNet» – <https://habr.com/ru/post/536692/>
 - Статья «XLNet против BERT» – <https://habr.com/ru/company/ods/blog/458928/>

«Потомки» модели BERT – DistilBERT, ALBERT, RoBERTa

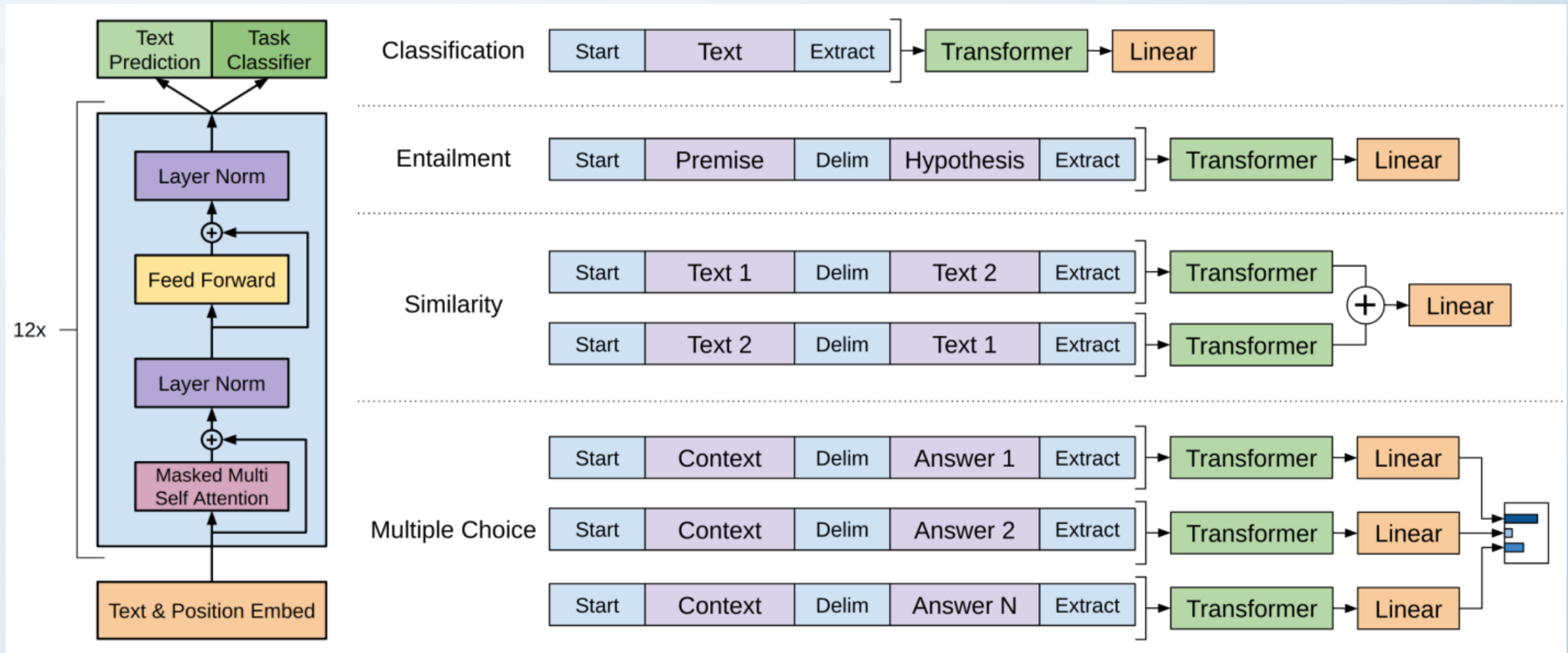
- Модель DistilBERT – «дистиллированная» модель BERT, предназначена для уменьшения размерности модели BERT и увеличения скорости вывода.
- Модель ALBERT является «облегченным» вариантом модели BERT. Как правило, показывает лучшее качество, чем BERT. Это достигается за счет оптимизации архитектуры нейронной сети – устранения избыточности (упрощения отдельных элементов архитектуры) и введения дополнительных связей между слоями.
- Модель RoBERTa показывает лучшее качество по сравнению с BERT по следующим причинам:
 - Размер обучающей выборки почти в 10 раз больше.
 - Маскирование осуществляется динамически в процессе обучения, что улучшает качество обучения.
 - Отказ от задачи предсказания следующего предложения и обучение на больших фрагментах документа.
- Источники:
 - Базовая статья «DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter» – <https://arxiv.org/abs/1910.01108>
 - «ALBERT: A Lite BERT for Self-supervised Learning of Language Representations» – <https://arxiv.org/abs/1909.11942>
 - Статья «ALBERT – облегченный BERT для самообучения языковым представлениям» – <https://habr.com/ru/post/524814/>
 - Базовая статья «RoBERTa: A Robustly Optimized BERT Pretraining Approach» – <https://arxiv.org/abs/1907.11692>
 - Статья «Evolving with BERT: Introduction to RoBERTa» – <https://medium.com/analytics-vidhya/evolving-with-bert-introduction-to-roberta-5174ec0e7c82>
 - Статья «BERT, RoBERTa, DistilBERT, XLNet – which one to use?» – <https://towardsdatascience.com/bert-roberta-distilbert-xlnet-which-one-to-use-3d5ab82ba5f8>

Модели GPT

- Группа языковых моделей, используемая для предсказания (продолжения) текста.
- Источники:
 - Базовые статьи:
 - [GPT-1](#)
 - [GPT-2](#)
 - [GPT-3](#)
 - Статья с прогнозами по модели GPT-4 – <https://towardsdatascience.com/gpt-4-is-coming-soon-heres-what-we-know-about-it-64db058cfd45>
 - Статья «The Journey of Open AI GPT models» - <https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2>
 - Статья с примерами использования GPT-3 - <https://vc.ru/ml/143516-kod-statya-verstka-i-muzyka-chto-uzhe-mozhet-sozdat-po-opisaniyu-samyy-moshchnyy-algoritm-generacii-tekstov-openai-gpt-3>
 - Статья с описанием работы GPT-3 - <https://habr.com/ru/post/514698/>

Модель GPT-1

- Появилась до BERT и решала задачи, схожие с BERT.
- Если BERT использует только кодировщики трансформеров, то GPT-1 использует только декодировщики.

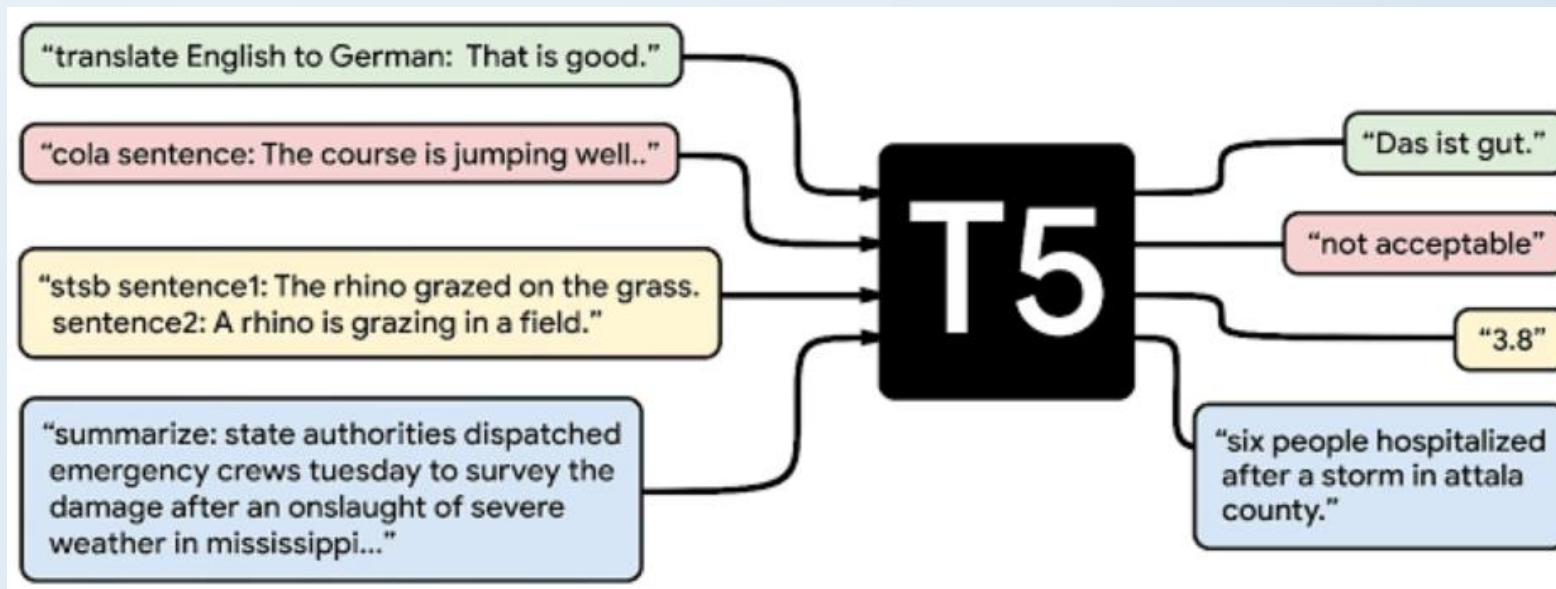


Модели GPT-2 и GPT-3

- Модель GPT-2 стала ответом на появление BERT.
- Основные изменения GPT-2 по сравнению с GPT-1:
 - Увеличение размера обучающей выборки.
 - Использование механизма Task Conditioning. Традиционно цель обучения модели определялась как $P(\text{output} | \text{input})$. В случае Task Conditioning цель определяется как $P(\text{output} | \text{input}, \text{task})$, то есть зависит от решаемой задачи.
 - Использование механизма Zero Shot Learning, при использовании которого цель применения модели формулируется на основе обучающей выборки.
- Основные изменения GPT-3 по сравнению с GPT-2:
 - Увеличение размера обучающей выборки и числа параметров модели, некоторые изменения в архитектуре сети.
- Статья со сравнением моделей - <https://dzone.com/articles/gpt-2-gpt2-vs-gpt-3-gpt3-the-openai-showdown>

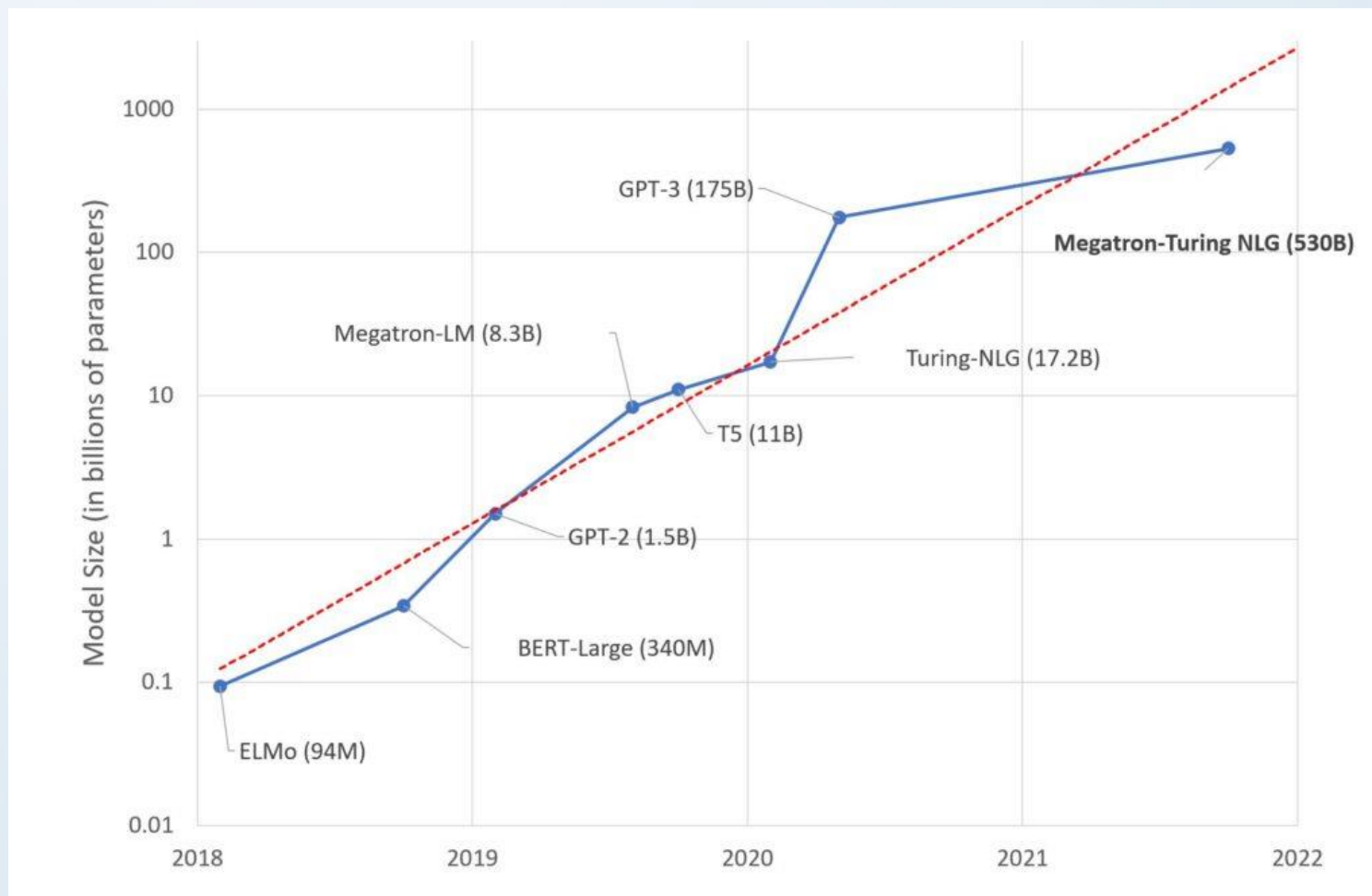
Модель Google T5

- Использует трансформерную нейросетевую архитектуру с небольшими улучшениями.
- Обучается на очень большом датасете.
- Позволяет решать различные задачи обработки ЕЯ.
- Является одной из наиболее практичных моделей. Доступна в Google Collab, опубликована в репозитории - <https://github.com/google-research/text-to-text-transfer-transformer>
- Источники:
 - Базовая статья «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer» – <https://arxiv.org/abs/1910.10683>
 - Блог компании Google - <https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html>
 - Статья - <https://towardsdatascience.com/t5-text-to-text-transfer-transformer-643f89e8905e>
 - Статья «Многозадачная модель T5 для русского языка» – <https://habr.com/ru/post/581932/>



Сверхбольшие модели – MT-NLG

- В дополнение к возможностям предыдущих моделей позволяет выполнять арифметические действия.
- Источники:
 - Базовая статья – <https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/>



Сверхбольшие модели – PaLM

- Используется стандартная трансформерная архитектура нейронной сети с рядом улучшений.
- Модель содержит 540 миллиардов параметров.
- Обучена с использованием 6144 тензорных процессоров Google TPU на платформе параллельных вычислений Pathways. Данные для обучения включали комбинации английских и многоязычных датасетов с текстами сайтов, книг, статей Википедии, чатов и кодов с GitHub.
- PaLM была проверена в нескольких сотнях задач по пониманию и генерации языка и показала превосходство в большинстве из них, причем во многих случаях – со значительным отрывом. Среди таких задач определение причин и следствий, понимание содержания, угадывание фильмов по эмодзи, поиск синонимов и контраргументов, перевод текста. В задачах ответы на вопросы и составления выводы точность PaLM превосходит аналогичные модели в несколько раз.
- Источники:
 - Базовая статья «PaLM: Scaling Language Modeling with Pathways» – <https://arxiv.org/abs/2204.02311>
 - Блог компании Google - <https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html>
 - Статья на русском языке - <https://neurohive.io/ru/papers/palm-yazykovaya-model-google-s-540-milliardami-parametrov/>

Explaining a Joke

Input: Did you see that Google just hired an eloquent whale for their TPU team? It showed them how to communicate between two different pods!

Model Output: TPUs are a type of computer chip that Google uses for deep learning. A "pod" is a group of TPUs. A "pod" is also a group of whales. The joke is that the whale is able to communicate between two groups of whales, but the speaker is pretending that the whale is able to communicate between two groups of TPUs.

Logical Inference

Input: Shelley is from Virginia, but is visiting that city with that famous market where they throw the fish. Going home next Tuesday!

Question: Is it likely that Shelley will be near the Pacific Ocean this weekend?

Model Output: The city with the famous market where they throw the fish is Seattle, Washington. Seattle is on the Pacific Ocean. Shelley is visiting Seattle, so she will be near the Pacific Ocean this weekend. The answer is "yes", it is likely that Shelley will be near the Pacific Ocean this weekend.

Выводы

- В настоящее время наблюдается тенденция к созданию сверхбольших языковых моделей.
- Сверхбольшие языковые модели способны только имитировать ИИ.
- Ян ЛеКун:
 - ...пытаться построить интеллектуальные машины путём масштабирования языковых моделей — всё равно что строить высотные самолёты для полёта на Луну. Вы можете побить рекорды высоты, но полёт на Луну потребует совершенно другого подхода.