**Лекция 10**

**Модель нейронной сети BERT**

Модель BERT (Bidirectional Encoder Representations from Transformers — “двунаправленные презентации кодировщика для трансформеров”) была представлена миру в статье, опубликованной исследователями из Google AI Language. Она вызвала нешуточный ажиотаж в сообществе машинного обучения, представив самые передовые на сегодняшний день результаты для целого ряда разных NLP (Natural Language Processing — “обработка естественного языка”) задач, включая формирование ответов на вопросы (SQuAD v1.1), формирование рассуждений на естественном языке (MNLI) и множество других категорий.

Ключевым техническим нововведением BERT является применение двунаправленного обучения трансформеров (популярной нынче модели с механизмом “внимания”) к языковому моделированию. Этот подход идет в разрез с предыдущими работами, которые рассматривали текстовую последовательность либо только слева направо, либо сочетали обучение слева направо и справа налево. Результаты этой работы показывают, что языковая модель с двунаправленным обучением способна достичь более глубокого понимания языкового контекста и потока, чем однонаправленные языковые модели. В статье исследователи подробно описывают новую технику под названием MLM (Masked Language Model — “маскированное языковое моделирование”), которая позволяет проводить двунаправленное обучение в моделях, для которых ранее это было невозможно.

**Предыстория**

Исследователи в области компьютерного зрения неоднократно демонстрировали пользу трансферного обучения (предварительного обучения модели нейронной сети на хорошо известной задаче, как например ImageNet, с последующим дообучением) с использованием уже обученной нейронной сети в качестве основы для новой модели с конкретной направленностью. В последние годы исследователи пришли к выводу, что подобная техника может быть чрезвычайно полезна и во многих задачах обработки естественного языка.

Еще один подход, проиллюстрирован в статье ELMo, который также очень популярен в NLP-задачах, — это обучение на основе признаков. В этом подходе предварительно обученная нейронная сеть создает векторные представления слов (word embeddings), которые затем используются в качестве признаков в NLP-моделях.

**Как работает BERT**

BERT использует трансформер — механизм “внимания”, который изучает контекстуальные отношения между словами (или подсловами) в тексте. В своей оригинальной форме трансформер включает в себя два отдельных механизма — кодировщик, который считывает введенный текст, и декодер, который выдает прогноз для задачи. Поскольку целью BERT является создание языковой модели, то ей необходим только кодировщик. Подробный разбор работы трансформера приведен в [статье](https://arxiv.org/pdf/1706.03762.pdf) Google.

В отличие от направленных моделей, которые считывают вводимый текст последовательно (слева направо или справа налево), кодировщик трансформера считывает сразу всю последовательность слов. Поэтому он считается двунаправленным, хотя правильнее было бы сказать, что он не имеет конкретного направления. Эта особенность позволяет модели изучать контекст слова на основе всего его окружения (слева и справа от слова).

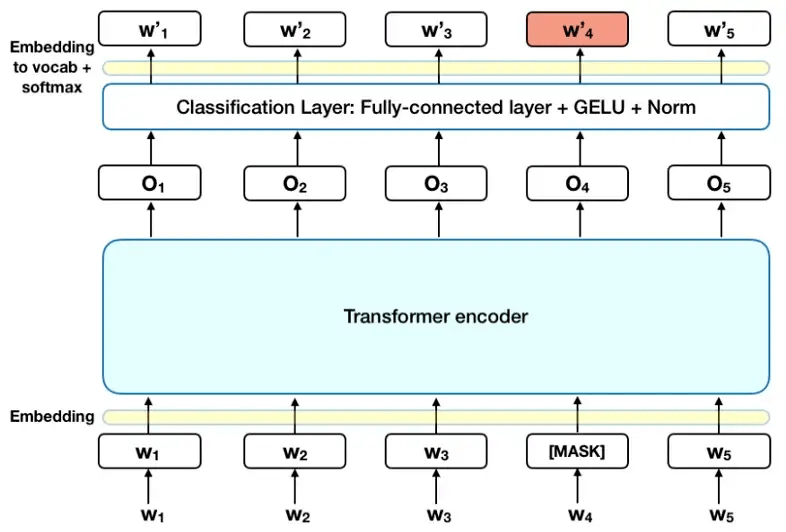
На приведенном ниже графике представлена общая структура кодировщика трансформера. Вход представляет собой последовательность токенов, которые сначала встраиваются в векторы, а затем обрабатываются в нейронной сети. Выход представляет собой последовательность векторов размера H, в которой каждый вектор соответствует входному токену с тем же индексом.

При обучении языковых моделей возникает проблема определения цели прогнозирования. Многие модели предсказывают следующее слово в последовательности (например, “Ребенок пришел домой из \_\_\_”) — это направленный подход, который по своей сути ограничивает контекстное обучение. Чтобы преодолеть эту проблему, BERT использует две стратегии обучения:

**Маскированное языковое моделирование (MLM)**

Перед вводом последовательности слов в BERT 15% слов в каждой последовательности заменяется токеном [MASK]. Затем модель пытается предсказать исходное значение замаскированных слов на основе контекста, предоставляемого другими, не замаскированными словами в последовательности. С технической точки зрения, предсказание выходных слов требует:

1. Добавления слоя классификации поверх выходных данных кодировщика.
2. Умножение выходных векторов на матрицу векторных представлений словаря (embedding matrix), преобразующую их к размерности словаря.
3. Расчет вероятности каждого слова в словаре с помощью softmax.



Функция потерь BERT учитывает только прогнозы замаскированных значений и игнорирует прогнозы не замаскированных слов. Как следствие, модель сходится медленнее, чем направленные модели, что компенсируется большей осведомленностью о контексте (смотрите Выводы #3).

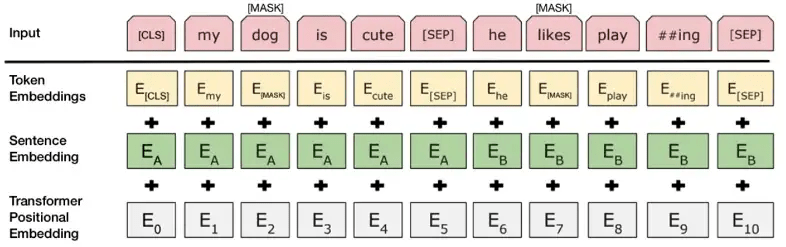
*Примечание: На практике реализация BERT немного сложнее и не заменяет все 15% замаскированных слов. Дополнительную информацию ищите в*[*Приложении A*](https://www.lyrn.ai/2018/11/07/explained-bert-state-of-the-art-language-model-for-nlp/#appendix-A)*.*

**Прогнозирование следующего предложения (NSP)**

В рамках процесса обучения BERT модель в качестве входных данных получает пары фраз, на которых она учится предсказывать, является ли вторая фраза в паре следующей после первой в исходном тексте. Во время обучения 50% входных данных представляют собой пары, в которых вторая фраза действительно является следующей фразой в исходном тексте, а в остальных 50% в качестве второй фразы выбирается случайная фраза из того же текста. Предполагается, что случайная фраза будет не связана по смыслу с первой фразой.

Чтобы помочь модели различить две фразы в процессе обучения, перед входом в модель входные данные обрабатываются следующим образом:

1. В начало первой фразы вставляется токен [CLS]. В конец каждой из фраз вставляется токен [SEP].
2. К каждому токену добавляется эмбеддинг (векторное представление) фразы, обозначающий Фразу A или Фразу B. Эмбеддинги фраз по своей концепции аналогичны эмбеддингам токенов со словарем из двух элементов.
3. К каждому токену добавляется позиционный эмбеддинг, чтобы указать его положение в последовательности. Концепция и реализация позиционного эмбеддинга хорошо раскрыты в статье, посвященной трансформеру.



*Источник:*[*BERT*](https://arxiv.org/abs/1810.04805)*[Devlin et al., 2018], с некоторыми изменениями*

Чтобы предсказать, действительно ли вторая фраза связана с первой, выполняются следующие шаги:

1. Вся входная последовательность проходит через модель-трансформер.
2. Выход токена [CLS] преобразуется в вектор размерности 2×1 с помощью простого слоя классификации (обученные матрицы весов и смещений).
3. Вычисление вероятности IsNextSequence с помощью softmax.

В процессе обучения BERT-модели, MLM и NSP обучаются вместе с целью минимизировать комбинированную функцию потерь двух стратегий.

**Как использовать BERT (дообучение)**

Использовать BERT для конкретной задачи относительно просто:

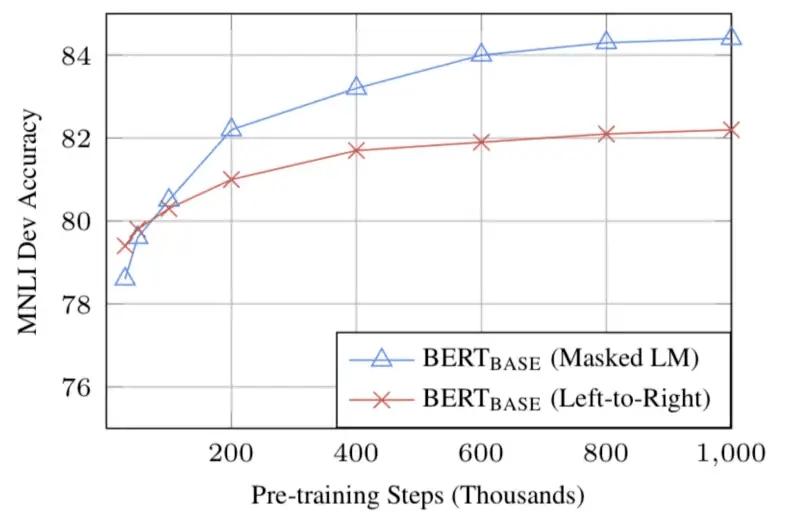
BERT можно использовать для самых разных языковых задач, добавляя небольшой для корректировки базовой модели:

1. Задачи классификации, такие как, например, анализ тональности, выполняются  аналогично классификации “следующей фразы”, добавляя слой классификации поверх выходных данных трансформера для токена [CLS].
2. В задачах формирования ответов на вопросы (например, SQuAD v1.1) программа получает вопрос относительно текстовой последовательности и должна отметить ответ в этой последовательности. Используя BERT, модель вопрос/ответ можно обучить, изучая два дополнительных вектора, которые отмечают начало и конец ответа.
3. При распознавании именованных объектов (NER - Named Entity Recognition) программа получает текстовую последовательность и должна помечать различные типы объектов (человек, организация, дата и т. д.), которые появляются в тексте. Используя BERT, NER-модель можно обучить, пропуская выходной вектор каждого токена через классификационный слой, который прогнозирует NER-метку.

При дообучении большинство гиперпараметров остаются такими же, как и при обучении базовой BERT. В статье даются конкретные рекомендации (раздел 3.5) по гиперпараметрам, требующим дообучения. Команда, разработавшая BERT, использовала эту технику для достижения самых высоких на сегодняшний день результатов в широком спектре сложных задач обработки естественного языка, подробно описанных в четвертом разделе статьи.

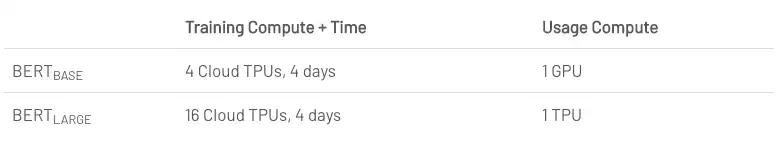
**Выводы**

1. *Размер модели имеет значение даже при большом масштабе.* BERT\_large с 345 миллионами параметров — самая большая модель в своем роде. Даже на небольших задачах очевидно превосходит BERT\_base, которая реализует ту же архитектуру с “всего” 110 миллионами параметров.
2. *При достаточном количестве данных для обучения больше шагов обучения == более высокая точность.* Например, в MNLI-задаче точность BERT\_base улучшается на 1,0% при обучении с 1 млн шагов (размер батча - 128 000 слов) по сравнению с 500 000 шагов для батча того же размера.
3. *Двунаправленный подход BERT (MLM) сходится медленнее, чем подходы слева направо* (поскольку в каждом батче прогнозируется только 15% слов), но двунаправленное обучение все-равно превосходит обучение слева направо после небольшого количества предварительных шагов.



*Источник:*[*BERT*](https://arxiv.org/abs/1810.04805)*[Devlin et al., 2018]*

**Вычислительные потребности (обучение и применение)**



**Заключение**

BERT, несомненно, является прорывом в использовании машинного обучения для обработки естественного языка. Тот факт, что она вполне доступна и позволяет быстро настраивать ее, вероятно, позволит найти широкий спектр практических применений в будущем. В этом посте мы попытались лишь описать основные идеи статьи, не утопая в технических деталях. Для тех, кто хочет более глубокого погружения в эту тему, мы настоятельно рекомендуем прочитать оригинал статьи и сопутствующие ей статьи, на которые в ней есть ссылки. Еще одной полезной ссылкой для вас будет [исходный код](https://github.com/google-research/bert) и BERT-модели, которые охватывают 103 языка и были щедро представлены миру исследовательской группой в качестве открытого исходного кода.