Московский государственный университет имени М. В. Ломоносова

Факультет Вычислительной Математики и Кибернетики
Кафедра Математических Методов Прогнозирования

Отчет по проектной работе

«Классификация текстов с помощью модели GPT2»

Математические методы обработки текстов

Выполнил:
студент 2 курса магистратуры 617 группы
Елистратов Семен Юрьевич

Москва, 2023

1 Введение

Современные языковые модели способны решать многие задачи обработки естественного языка с высоким уровнем качества. Одной из таких моделей является GPT2 - часть семейства авторегрессионных генеративных моделей с архитектурой типа трансформер. Одной из главных причин широкого использования языковых моделей является transfer learning - возможность дообучения модели, предобученной на огромном наборе данных, для решения конкретной задачи. В рамках данного проекта проведен анализ возможности использования модели генерации текста GPT2 для решения задачи классификации с помощью transfer learning и проведено сравнение с другими моделями классификации.

2 Постановка задачи

2.1 Данные

В рамках данного проекта будет использован следующий набор данных для задачи классификации: ImdbMovieDataset (бинарная классификация). Данные представляют собой отзывы пользователей о фильмах. Задача sentiment analysis - классификация текста по эмоциями (позитивный/негативный отзыв в случае бинарной классификации).

Для каждого текста проводится предобработка и токенизация, после чего модель языковая модель преобразует токены в емебеддинги.

2.2 BERT

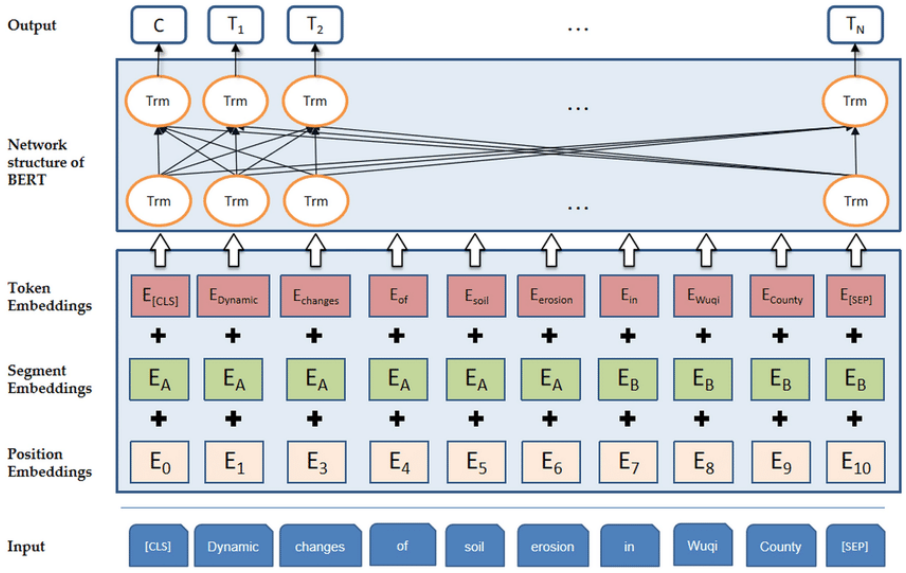

Рис. 1: Архитектура модели BERT

В качестве бейзлайна будет использована модель BERT. BERT(Bidirectional Encoder Representations from Transformers) имеет двунаправленную архитектуру типа трансформер. В процессе обучения она получает текст целиком и строит внутреннее представление сразу по всему контексту, максимизируя правдоподобие данного предложения. Использование двунаправленной архитектуры позволяет модели глубоко оценивать контекст предложения, что идеально соответствует задаче классификации текста. Для этого к модели будет добавлен слой классификации, который будет по последнему внутреннему представлению выдавать результат классификации.

2.3 GPT2

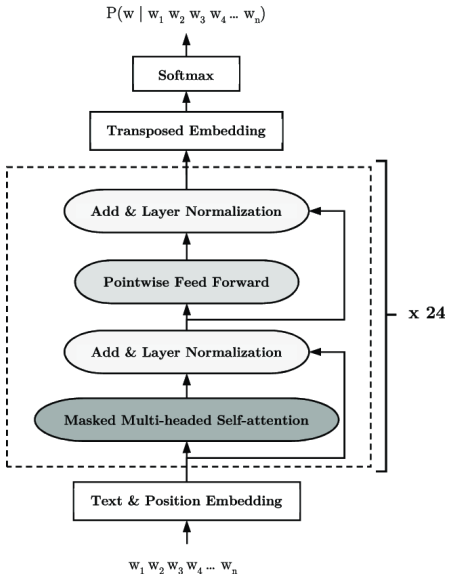

Рис. 2: Архитектура модели GPT2

GPT2 - авторегрессионная генеративная модель. В отличие от BERT она имеет однонаправленную архитектуру, таким образом она в первую очередь предназначена для генерации текста: предсказывание следующего токена по предыдущим. Однако в рамках данной работы проведены эксперименты по использованию данной модели для задачи классификации. Для этого предлагается два различных метода:

1. Добавление головы классификации к последнему/аггрегированному внутреннему представлению модели.
2. Предсказание класса через вероятности генерации текста.

Первый вариант аналогичен решению для BERT. Второй подход предполагает следующее: нам необходимо моделировать вероятность $p(class|text)$. Воспользуемся теоремой Байеса:

$$p(class|text) = \frac{p(text|class)p(class)}{p(text)}, \tag{1}$$

где $p(class)$ - константа. Таким образом, необходимо научить модель оценивать условную генерацию текста, что даст возможность оценивать $p(text|class)$. Оценивать $p(text)$ модель может изначально.

Тогда в процессе обучения нужно максимизировать:

$$\frac{p(text|class)p(class)}{p(text)} \to \max \equiv -\log p(text|class) - (-\log p(text)) \to \min$$

. Однако при прямом подходе к этой задаче оптимизации модель не сойдется к оптимальному качеству, так как будет сильно занижать вероятность $p(text)$. Поэтому простейший вариант - просто максимизировать $p(text|class)$. Кроме того, было проведена попытка максимизировать $\frac{p(text|class)}{p(text|!class)}$, которая показала плохое качество.

3 Эксперименты

Для начала был обучен бейзлайн и первый подход к классификации с GPT2. Для этого проведен перебор базовых гиперпараметров: размер батча, lr, weight decay. Кроме того, было проведено сравнение архитектур: в одном варианте голова для классификации получает последнее внутреннее представление, а в другом среднее внутренних представлений для всех токенов.

Модели обучались 4 эпохи (что достаточно для дообучения больших языковых моделей) с отслеживанием метрик на обучении и валидации.

Второй подход был реализован следующим образом: к исходному текстовому описанию в начало добавлялось два токена: номер класса и специальный токен <SEP>. Модель получала на вход данный текст, выдавала логиты для каждого токена, после чего вычислялась кросс-энтропия с исходным текстом без учета токена класса, <SEP> и паддинга. Таким образом, модель учитывала информацию о классе текста, однако лосс оценивал только вероятность генерации исходного текста.

Также, были проведены эксперименты с обучением разности итогового лосса с лоссом без информации о классе текста или с лоссом для текста, в котором первый токен был противоположный класс. Однако в таких подходах модель быстро завышала вычитаемый лосс в результате чего модель начинала хуже обучаться.

Для того, чтобы предсказать итоговый класс в исходный текст-описание добавлялся токен класса и для всевозможных классов вычислялись вероятности. После чего выбирался класс с максимальной вероятностью. Минус такого подхода, что необходимо прогонять модель для каждого класса в отдельности.

Результаты экспериментов:

	BERT/DistilBERT	GPT2	GPT2_gen
last agg	0.86208	0.86764	0.7667 (no agg)
mean agg	0.86776	0.86172	-

Таблица 1: Результаты сравнения моделей

Таким образом, можно сделать вывод, что дообучение GPT2 для классификации путем добавления головы работает не хуже, чем такой же метод для BERT. Следовательно, даже однонаправленная архитектура GPT2 способна правильно выделять контекст всего текста для определения его семантических свойств. Подход оценки вероятности через генерацию текста показал себя хуже, так как такую модель значительно сложнее правильно обучить.

4 Выводы

Модель GPT2 решает задачу классификации не хуже, чем BERT, несмотря на то, что последний лучше учитывает контекст из-за двунаправленной архитектуры. Таким образом, GPT2 также способна решать задачу классификации на приемлимом уровне путем добавление линейных слоев к последнему внутреннему представлению. Подход с предсказанием класса, через вероятность генерации текста является довольно интересным, однако показывает худшее качество из-за сложности обучения модели. Кроме того, он менее эффективен по времени относительно подхода с добавлением классификационной головы.