Федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский университет
«Высшая школа экономики»

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

КУРСОВАЯ РАБОТА

Исследовательский проект на тему

«Применение предобученной модели GPT в задачах обработки текстов»

Выполнила студентка группы 182, 3 курса,
Сусла Диана Михайловна

Руководитель КР:
приглашенный преподаватель, Симагин Денис Андреевич

Москва 2021

Содержание

1 Аннотация

В задачах распознавания текстов наблюдается значительный прогресс, что во многом обусловлено двумя вещами: появлением модели трансформер, а также возможностью её предобучения на больших корпусах текстов без разметки. В последствии полученная модель уже может быть специализирована для решения различных задач обработки текстов.

В рамках данного проекта предстоит провести эксперимент с предобученной моделью GPT для решения задач суммаризации и классификации текстов.

2 Abstract

There is significant progress in the text recognition tasks, which is largely due to two things: the introduction of the transformer model and the possibility of its pre-training on large text bodies. The resulting model can be specialized for solving various text processing tasks.

In this project, we will make an experiment with a pre-trained GPT model to solve the problems of text summarization and classification.

3 Список ключевых слов

Машинное обучение, глубинное обучение, векторное представление слов, NLP, трансформер, GPT, суммаризация текста, классификация текста.

4 Введение

4.1 Описание предметной области

На текущий момент нейронные сети используются в самых разных областях человеческой жизни. Одним из наиболее крупных направлений глубинного обучения является обработка текстов на естественном языке (NLP). Можно выделить следующий ряд задач, решаемых нейронными сетями в данной области: машинный перевод, генерация текстов, классификация предложений или текстов, пересказ текстов, поиск ответа на вопрос по тексту, чат-боты. Применение глубинного обучения в решении этих задач помогает значительно автоматизировать множество процессов.

4.2 Цель и задачи

Модель Generative Pretrained Transformer (GPT), как и другие модели, в основе которых лежит трансформер, сейчас активно применяются для решения самых различных задач обработки естественного языка (NLP) – от генерации текстов и до создания чат-ботов. В данной же работе будет рассмотрено применение модели GPT-2 для решения задач сжатого пересказа и классификации рецензий на фильмы. Зачастую, при выборе фильма или сериала людям хочется узнать об основных преимуществах и недостатках фильма. Но многие рецензии, написанные людьми, очень объемные, а в итоге содержат мало полезной информации. Данная работа направлена на решение этой проблемы. При прочтении сжатого пересказа рецензии пользователь сможет вынести для себя общую информацию о фильме и не тратить свое время на прочтении всей рецензии.

Таким образом основной целью проекта является анализ применения модели GPT-2 для решения задачи суммаризации и классификации текстов, в частности рецензий.

Говоря о задачах, конкретно можно выделить следующие: (1) изучение

необходимой теории в области глубинного обучения и обзор текущего состояния области, (2) примененние предобученной модели GPT-2 для решения задачи пересказа, (3) оценка качества полученных результатов, (4) сравнение результатов с теми, что уже имеются в других работах по суммаризации текстов.

5 Основная часть

5.1 Обзор литературы

Последнее десятилетие глубинное обучение развивалось очень стремительно. До сих пор с каждым годом создаются новые методы, улучшающие работу тех или иных моделей. Поэтому для того, чтобы иметь качественное представление об области необходимо разобраться с моделями, предшествующими тем, что активно используются сейчас, проследить за эволюцией методов.

5.1.1 Нейронные сети. Свёрточные нейронные сети

Наиболее простой нейронной сетью является многослойный перцептрон. Зачастую, уже он дает весьма неплохое качество и используется для решения ряда простых задач. Для работы с изображениями обычно используются свёрточные нейронные сети (CNN). По сравнению с полносвязными нейронными сетями свёрточные имеют значительно меньше параметров (из-за чего также менее склонны к переобучению), а также инварианты к небольшим преобразованиям, что как раз необходимо при работе с изображениями.

По сравнению с моделями машинного обучения нейронные сети дают хорошие результаты при работе с более сложными данными (например, с нечёткой структурой). А также решают ряд задач, которые классическим методам машинного обучения не под силу. Например, генерация видео и изображений, машинный перевод. Однако у нейронных сетей есть ряд особенностей. Для

эффективного решения задачи нужно тщательно подбирать архитектуру сети, обучение нейронной сети на большом корпусе данных требует больших вычислительных ресурсов. Также из-за большого числа параметров нейронные сети склонны к переобучению. Существует ряд методов, направленных на решение этой проблемы. Например, регуляризация или ранняя остановка. Но одним из наиболее популярных методов является Dropout [1]. Данный метод основан на обнулении коэффициентов у некоторых нейронов, тем самым упрощая модель (используя меньшее число параметров для предсказания) и не давая модели переобучиться. Также важным методом борьбы с переобучением является Data Augmentation [2]. Еще одной проблемой при обучении нейронных сетей является исчезновение градиента (vanishing gradient problem), решать которую пытается Batch Normalization, путём нормализации распределения нейронов [3]. Данный метод также ускоряет сходимость сети и уменьшает её чувствительность к изначальной инициализации. В свёрточных сетях также часто применяется Pooling [4]. Он тоже помогает ускорять работу нейронной сети путем уменьшения числа параметров, а также нужен для поддержания иерархичности.

5.1.2 Векторные представления слов

И наконец можем перейти к центральной теме данной работы, а именно к NLP. На данный момент существует множество методов работы с текстами. В основе большинства из них лежит представление картинок или элементов естественного языка в виде некоторого вектора чисел. Говоря о применении векторных представлений в NLP, в целом идея построения векторных представлений основана на том, что контекст слова тесно связан с самим словом. Самой популярной на данный момент моделью на основе векторных представлений слов является Word2Vec [5]. Модель имеет ряд преимуществ. Например, она не требует большой вычислительной мощности, не нуждается в предобработке данных, а также имеет высокую скорость обучения. Но вместе с этим Word2Vec имеет свои недостатки. Модель плохо работает

на длинных текстах, не может представить слова, не встречавшиеся в обучающей выборке. А главной проблемой является то, что она не учитывает семантически отношения между словами. Последнюю проблему решает модель Global Vectors (GloVe), путем использования матрицы совпадений [6]. Основным преимуществом GloVe становится то, что получаемые векторные представления выходят более осмысленными чем у Word2Vec, что помогает более качественно решать поставленную задачу. Оба данных метода показывают очень хорошие результаты, однако они не учитывают морфологическое богатство языка. Для решения этой проблемы используется модель FastText, которая по сути является модификацией Word2Vec [7].

5.1.3 Рекуррентные нейронные сети

Еще одним видом нейронных сетей являются рекуррентные нейронные сети (RNN). Данный вид сетей активно применяется для решения задач NLP. Однако основная проблема RNN в том, что они не могут обучаться долгосрочным зависимостям, то есть учитывают только ближайший контекст. Данная проблема решается в Long Short-term Memory (LSTM) – наиболее часто используемом на данный момент типе рекуррентных нейронных сетей [8]. Также LSTM лежит в основе довльно сильной модели для решения задачи NLP – ELMo [9]. ELMo по сути представляет собой две LSTM направленных в разные стороны, благодаря чему векторные представления слов в данной модели становятся контекстно-зависимыми, что помогает более качественно решать поставленную задачу.

5.1.4 Трансформеры

Рекуррентные нейронные сети также используются вместе с алгоритмом внимания (RNN seq2seq with attention) [10]. Внимание добавляется для того, чтобы учитывать контекст слова в исходном предложении и выделять его наиболее важную часть. Однако у такой модели есть несколько недостатков: нейронная сеть работает очень долго и процесс работы плохо параллелится.

Решить эту проблему помогает довольно новая и широко используемая сейчас для решения задач NLP модель – трансформер. Модель трансформера состоит из двух основных частей – декодера и энкодера.

Рис. 5.1: Блок энкодера трасформера.

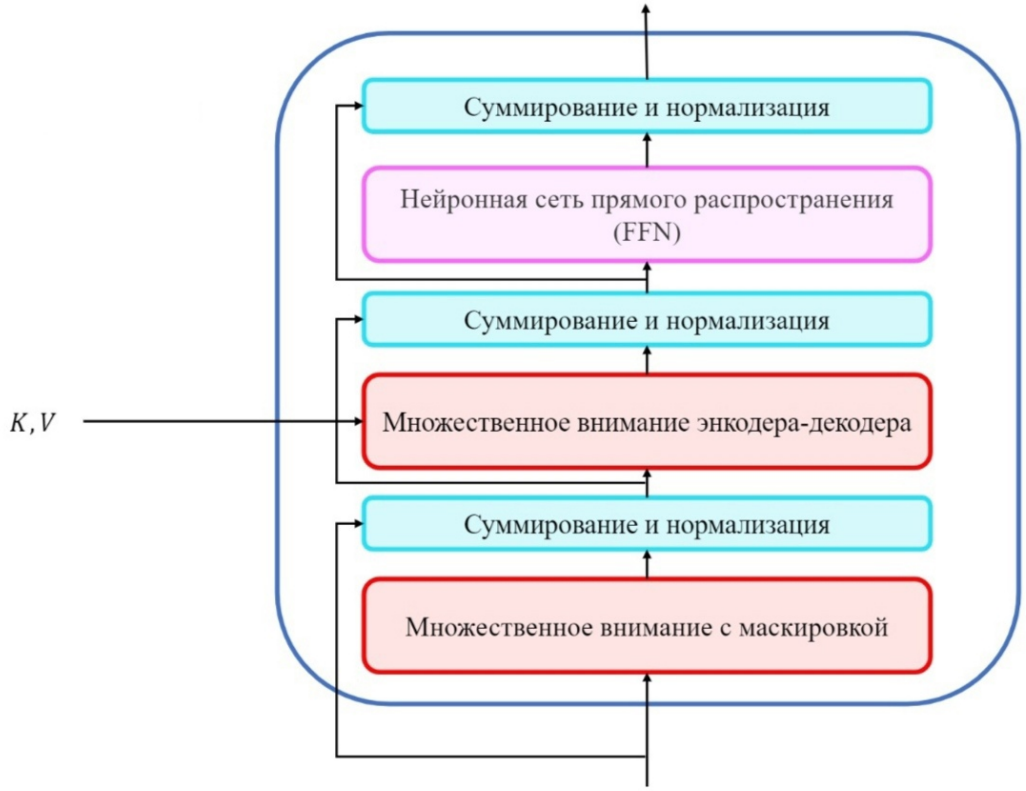

Рис. 5.2: Блок декодера трансформера.

На рисунках 5.1 и 5.2 можно видеть строение каждого из блоков трансформера.

В классических seq2seq со вниманием мы применяем внимание только в декодере, в трансформерах же оно применяется и в энкодере, поскольку было бы полезно при обновлении представления текущего токена учитывать контекст токенов вокруг. В целом внимание в трансформерах устроено чуть сложнее. У каждого представления есть три представления, которые используются в механизме внимания:

Query: вектор, который используется для запроса. Во внимании до этого роль Query выполняло само представление токена, для которого сейчас вычисляется внимание.

Key: вектор, который используется при обращении. Во внимании до этого роль Key выполняло само представление токена, который использовался для вычисления внимания для другого токена Value: вектор, который фигурирует в взвешенной сумме.

Аналогично используется внутреннее внимание и в декодере, однако при обучении нам доступен весь перевод целиком, декодер же не должен уметь смотреть на будущие токены, которые он должен предсказать, поэтому мы должны маскировать их, это называется внутренне внимание с маскировкой (Masked Self-attention). Маскировка происходит путём замены скалярного произведения на $-\infty$ чтобы значения выхода Softmax для будущего токена было близко к 0, и во взвешенной сумме он не вносил вклад.

Трансформер, например, лежит в основе такой модели как Bidirectional Encoder Representations from Transformers (BERT) [11]. По сути BERT представляет собой стэк слоев энкодера из трансформера. Обучается BERT на две основные цели – masked language modeling (предсказание пропущенного слова в предложении) и next sentence prediction (определение того, являются ли предложения последовательными). Еще одной моделью, в основе которой лежит трансформер, является Generative Pre-trained Transformer (GPT) [12]. Основное отличие GPT от BERT в том, что в BERT используется полный

self-attention, который рассматривает контекст слова в рамках всего предложения, а в GPT используются внимание слева направо, то есть контекст учитывается по словам, стоящим в предложении левее рассматриваемого слова. Обе модели показывают отличные результаты при решении задач NLP и зачастую выбор конкретной модели зависит от специфики задачи.

Стоит подробнее рассмотреть модель, с помощью которой и будет производиться решение поставленной задачи в рамках данного проекта, а именно – GPT-2 [13]. По сути GPT-2 является улучшенной модификацией предшествующей ей модели GPT-1, обученной на большем корпусе текстов. GPT-2 имеет четыре основные вариации – маленькая, средняя большая и очень большая, которые отличаются между собой количеством параметров, а следовательно размером сети. В рамках данного проекта будут рассмотрены маленькая и средняя версия, содержащие 117 и 345 миллионов гиперпараметров соответственно.

В отличие от BERT GPT-2 представляет собой стэк слоев декодера трансформера и обучена предсказывать следующее слово в предложении. Входной токен обрабатывается всеми слоями декодера и преобразуется в вектор. Результатом произведения этого вектора на матрицу векторных представлений слов будет коэффициент для каждого слова из словаря. Чтобы лучше предсказывать слова, GPT-2 опирается на контекст с помощью внутреннего внимания – модель оценивает значимость каждого слова в сегменте текста. В конце очередного этапа модель рассматривает не одно слово с наибольшим коэффициентом, а некоторое множество слов, из которых выбирается наиболее подходящее по контексту. Полученный выход подается во входную последовательность следующего шага, на основе которой предсказывается следующее слово. Что делает модель GPT-2 авторегрессионной.

В целом модель GPT-2 активно используется для следующих целей: (1) генерация текста, (2) краткий пересказ текста или обобщение, (3) ответы на вопросы исходя из содержания текста, (4) машинный перевод текста и задач, являющихся производными от данных, например, создание чат-ботов.

Стоит также упомянуть о более новой модели GPT-3 [14]. GPT-3 является модификацией модели GPT-2, но имеет большее контекстное окно (2048 токенов против 1024 в GPT-2), при этом используя меньшее число слоев при равном количестве параметров в сети. Также в модели используется разреженное внимание, которое смотрит не на все токены, а на какие-то фиксированные паттерны. На данный момент GPT-3 является самой большой моделью из существующих. Она обучена на корпусе текстов размером 570 гигабайтов и имеет 175 миллиардов параметров. Благодаря данным фактам и удачной архитектуре, хорошо моделирующей естественный язык, модель способна решать новые задачи практически без дообучения на новых данных, что значительно отличает ее от всех предшествующих моделей.

5.2 Анализ состояния области

Теперь обсудим центральную задачу данной работы – применение GPT-2 для суммаризации текстов. Существует два основных подхода к сжатию текстов [15] – экстрактивный и абстрактивный. Экстрактивный подход заключается в извлечении из исходного текста наиболее «значимых» информационных блоков. В качестве блока могут выступать отдельные абзацы, предложения или ключевые слова. Абстрактивный же подход заключается в генерации краткого содержания с порождением нового текста, содержательно обобщающего исходный текст. На практике экстрактивный подход хоть и поддерживает разумную степень грамотности и точности текста, но генерирует тексты, читаемость и связность которых хуже, чем у написанных человеком. Абстрактивный же, хоть и является более сложным, используя семантическое представление текста, выдает более близкие к человеческому пересказу результаты. В нашем случае будет сделан упор на экстрактивный метод суммаризации. Так как абстрактивный подход разумнее применять для пересказа больших текстов, где выделенные предложения или слова смогут корректно передать всю суть текста.

На данный момент state-of-the-art моделью для решения задачи суммаризации текстов (а если быть конкретным, абстрактивной суммаризации) является Bidirectional Autoencoder Representations from Transformers (BART) [16]. BART комбинирует в себе характеристики из BERT и GPT-2: энкодер из BERT, а декодер из GPT-2. В целом архитектура BERT из-за наличия полного self-attention больше подходит для решения задач пересказа текстов. А GPT-2 принято считать, что больше подходит для решения генеративных задач и является менее эффективной, когда вся входная последовательность может использоваться для генерации выходного токена. В данном же проекте модель GPT-2 будет рассмотрена как раз для решения задачи суммаризации текстов, для того, чтобы сравнить полученные результаты с уже имеющимися при использовании других моделей, а также имеющимися применениями GPT-2 для сжатия текстов.

Существует некоторое количество работ о применении BERT, GPT-2 и других моделей для решения задачи суммаризации текстов. В статье [15] , уже упомянутой выше, описана модель BERTSUM. Данная модель была обучена на задачу экстрактивной суммаризации текста при помощи модификации исходной модели BERT. BERTSUM обучен на классическом для данной задаче наборе данных – «CNN/Daily News». По сути модель выполняет задачу бинарной классификации для того, тобы предсказать, будет ли каждое конкретное предложение включено в пересказ или нет. Однако, так как изначально BERT не создавался в качестве модели для генерации текстов, его использование для задачи суммаризации все-таки ограничено.

Идея абстрактивной суммаризации текста была реализована в статье [17], где для этой цели использовалась модель Sequence-to-Sequence (seq2seq), основанная на архитектуре кодировщика и декодировщика. В данной модели кодировщик считывает исходный текст, преобразует его в некие скрытые состояния, а декодировщик принимает скрытые состояния и преобразует их в итоговый пересказ.

Также можно рассмотреть работу, исследующую применение BERT и GPT-

2 для суммаризации медицинских статей о COVID-19 [18]. Данная работа направлена помочь ученым и исследователям в сфере здравоохранения получать данные о вирусе, для поиска методов борьбы с ним. Говоря о технической состоявляющей исследования, авторы опробовали как абстрактивный, так и экстрактивный методы суммаризации. Экстрактивная суммаризация была реализована с помощью метода k-средних с последующим поиском k ближайших соседей. Таким образом производился поиск предложений, хорошо отражающих смысловое содержание текста.

Если же говорить о работах, в которых непосредственно для решения задачи суммаризации использовались рассматриваемые нами данные, в качестве примера можно рассмотреть работу [19]. В работе реализовалась абстрактивная суммаризация с использованием достаточно несложной по структуре нейронной сети. Декодер представляет собой два слоя LSTM с использованием dropout. В декодере же вместо обычного внимания используется Bhadanau attention. Для обучения модели были выбранны следующий параметры: размер батча = 64, число эпох обучения = 7, обучающая способность = 0.005. А само обучение проходило на подмножестве данных размером 50000. Результаты такого обучения модели сложно оценить, так как в работе не используется никакая стандартная метрика. Автор решил ограничиться проверкой текстов на связность человеком. В целом, полученные краткие содержания вышли в основном связными и отражающими суть рецензий.

При решении задачи пересказа текста, как и в целом при решении задач машинного перевода, генерации текстов и других, возникает естественная необходимость в оценке качества полученного текста. Существует немало подходов, но в рамках данной работы нами будет рассмотрено два метода – Bilingual Evaluation Understudy (BLEU) [20] и Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [21]. Говоря об общем понимании методов: BLEU определяет точность, то есть сколько слов из сгенерированного моделью текста совпадает с теми, что есть в тексте, написанном человеком. А ROUGE определяет полноту, то есть сколько слов из текста, написанного человеком,

совпадает с теми, что есть в тексте, сгенерированном моделью. Основной проблемой при оценке качества сгенерированного текста является невозможность оценить связность при помощи каких-либо автоматических методов. И зачастую приходится проверять качество, используя человеческие ресурсы.

С классификацией же дела обстоят лучше. Существует немало различных подходов к решению задачи классификации. Например, часто для её решения используются модели классического машинного обучения. Во многих случаях уже с помощью них можно достигнуть хорошего качества, а иногда и более высокого, чем с помощью нейронных сетей. Например в работе [22]ля решения задачи многоклассовой классификации используется модель на основе SVM, а если конкретно, структурированного SVM – модифицированной версии стандартного алгоритма. В то время как классификатор SVM поддерживает бинарную классификацию, многоклассовую классификацию и регрессию, структурированный SVM позволяет обучать классификатор для общих структурированных выходных меток. Для обучения использовался небольшой набор данных из 5000 экземпляров. В итоге моделе удалось достичь значения accuracy около 85.4%, что явлется отличным результатом для многоклассовой классификации.

Также существует немало работ, в которых решалась задача классификации при помощи нейронных сетей. В некоторых работах описано применение различных моделей машинного и глубинного обучения для многоклассовой классификации. Например, в работе [23]ечь идет о настройке и дообучении модели BERT для решения задачи классификации текстов. В ней задача классификации решалась не только при помощи дообучения модели BERT, но также при параллельном применении multitask-learning.

Но в основном, если говорить о наборах данных, содержащих рецензии, рассматривается задача бинарной классификации, так называемый sentiment analysis. В том числе имеются решения, в которых также рассматривались наборы данных с отзывами, например, [24]. В ней использовался небольшой набор данных «Stanford movie review». В качестве функции потерь исполь-

зовалась кросс-энтропия. Автору удалось достичь значения функции потерь равного 0.057, что является неплохим результатом с учетом того, что обучающий набор данных был крайне мал.

Также задача бинарной классификации решается в работе [25]. В качестве базовой модели используется BERT и также подход, применяемый уже после обучения модели. Когда обучение происходит только на текстах рецензий, а уже после этого модель настраивается на нескольких итоговый задачах, то есть в нашем случае пересказах текстов. Лучшее достигнутое значение accuracy оказалось равно 78.07.

5.3 Ход и результаты работы

В рамках данного проекта для решения задач классификации и суммаризации текстов была использованна предобученная на 40 Гб текстов (WebText) модель GPT-2. Для решения поставленной задачи выбрана именно эта модель, так как она является оптимальной в рамках тех условий, которые имеются для выполнения студенческого проекта. Модель достаточно мощная и хорошо справляется с различными задачами NLP, но при этом её маленькая и средняя версии содержат намного меньше параметров, чем GPT-3, содержащая 175 миллиардов параметров, что позволяет обучать их, используя доступные графические процессоры (GPU) с приемлемыми мощностями. Дообучение модели происходило в среде Google Collab с использованием имеющейся GPU с объемом памяти 16 ГБ.

Для дообучения модели было расмотрено два схожих по своей сути набора данных – «Amazon Fine Food Reviews» и «Amazon Review Data (2018). Movies and TV». Первый набор данных помимо самих текстов рецензий, оценок (числа от 1 до 5) и других признаков, не рассматриваемых нами, содержит название рецензии, которое по своей сути представляет собой очень краткое содержание текста. Второй же набор данных имеет большое число признаков, среди которых – текст рецензии, оценка (число от 1 до 5) и крат-

кая рецензия на фильм, так называемое «summary». Для решения задачи суммаризации было решено рассмотреть в качестве таргета название отзыва в случае первого набора данных и краткий отзыв в случае второго.

Первым этапом решения обеих поставленных задач являлась базовая предобработка данных. Во-первых, модель GPT-2 может обработать 1024 токена. Каждый токен проходит через все блоки декодера по своей собственной траектории. Так что для того, чтобы далее не возникало никаких проблем с работой моделей, из наборов данных были исключены содержащие более 1024 токенов. Также при проведении анализа датасетов было выяснено, что в них содержатся как весьма содержательные рецензии большого объема, так и рецензии небольшого объема, дальнейшая работа с которыми не принесет хороших результатов. Очевидно, что не имеет смысла суммаризировать слишком короткие рецензии, так что в датасете были оставлены все рецензии, длиной более 80 символов. В итоге из 568454 имеющихся в наборе данных «Amazon Fine Food Reviews» текстов после фильтрации осталось 567702. А в наборе «Amazon Review Data (2018). Movies and TV» – 210954 из 500000.

После этого была проведена предобработка текстов рецензий и пересказов. В целом предобработка производится с целью исключения различного рода факторов, снижающих качество данных и мешающих работе моделе. При помощи регулярных выражений тексты были преобразованы в более приемлемый для работы формат. Были удалены все символы, не являющиеся словами, ссылки и т.д., удалены стоп-слова (артикли, междометия, союзы и другие) при помощи блока *stopwords* из библиотеки *nltk*. Также с использованием общедоступного списка сокращений на английском языке сокращения были преобразованы в полные формы, а также слова в текстах были приведены к нижнему регистру.

5.3.1 Классификация

Как уже было упомянуто ранее, для решения задачи классификации было решено рассмотреть два схожих датасета, первый - с отзывами на фильмы и

сериалы, второй - с отзывами на еду. Интересно понять, как сильно качество классификации зависит от тематики рецензий. Есть ли вообще корреляция и если есть, то в пользу какого из наборов данных.

После предобработки данных был проведен их анализ. В частности, анализ рапределения оценок.

Рис. 5.3: Распределение оценок в первом наборе данных (отзывы на фильмы)

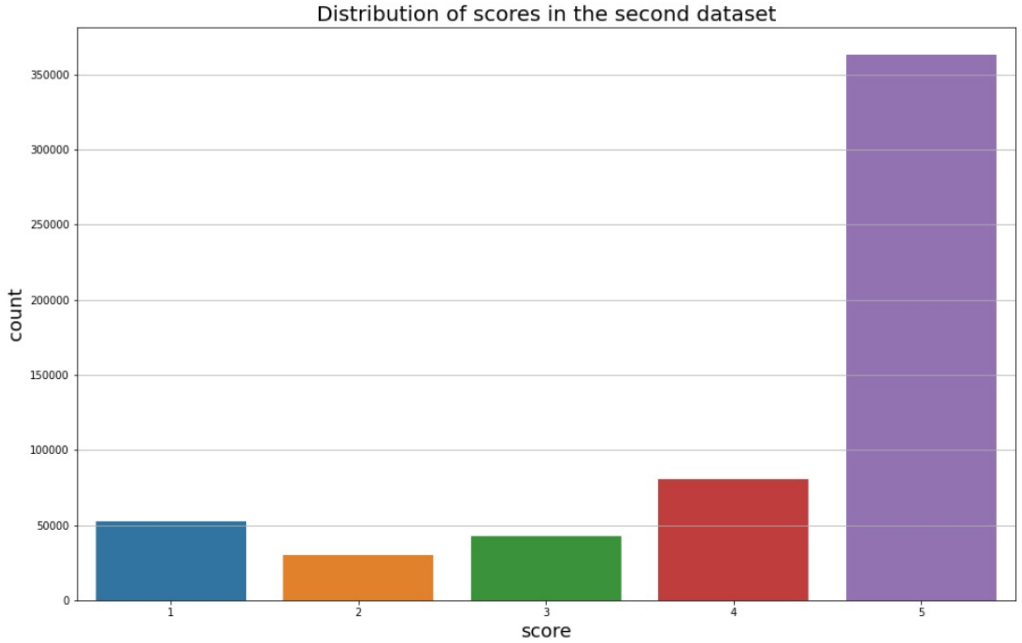

Рис. 5.4: Распределение оценок во втором наборе данных (отзывы на еду)

Как можно заметить на рисунках 5.3 и 5.4 рассматриваемые нами наборы данных имеют явную несбалансированность классов – оценок "5" боль-

ше, чем суммарно всех остальных оценок, так что оценивать качество работы исключительно при помощи подсчёта accuracy было бы не совсем корректно. Для решения этой проблемы мною было рассмотрено два подхода к оценке качества многоклассовой классификации – микро-усреднение и макро-усреднение. Для того, чтобы разобраться, как работает каждый из них, введем некоторые определения.

Рассмотрим на примере бинарной классификации, для многоклассовой всё определяется аналогичным образом. Если y – истинная метка класса объекта, a – рассматриваемая модель, а $a(x)$ ответ модели для входа x, то TP, FP, FN и TN определяются следующим образом (рисунок 5.5):

	$y = 1$	$y = -1$
$a(x) = 1$	True Positive (TP)	False Positive (FP)
$a(x) = -1$	False negative (FN)	True Negative (TN)

Рис. 5.5: Матрица ошибок

При микро-усреднении TP, FP, FN и TN сначала усредняются по всем классам, а затем вычисляется итоговая метрика. А при макро-усреднении сначала вычисляется итоговая метрика для каждого класса, а затем уже результат усредняется по всем классам. Если какой-то класс имеет очень маленькую мощность, то при микро-усреднении он практически никак не будет влиять на результат, поскольку его вклад в средние TP, FP, FN и TN будет незначителен. В случае же с макро-вариантом усреднение проводится для величин, которые уже не чувствительны к соотношению размеров классов (если мы используем, например, точность или полноту), и поэтому каждый класс внесет равный вклад в итоговую метрику.

В качестве метрики оценки качества классификации были рассмотрены следующие: accuracy, precision, recall и F1 мера. Посчитать каждую из них можно по формулам, представленным ниже.

$$accuracy = \frac{TP + TN}{TP + FN + TN + FP}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{recall \cdot precision}{recall + precision}$$

Обучение моделей происходило в два этапа. На первом для каждого из наборов данных были использованы как классификатор на базе модели BERT, так и классификатор на основе модели GPT-2. Было обучено пару эпох и по результатам обучения было принято решение далее использовать только модель GPT-2, которая дала более высокое качество. На втором этапе на каждом датасете была дообучена модель GPT-2 на 4 эпохах со скоростью обучения = 2e-5 и размером батча = 32. Также для обучения модели был использован оптимизатор AdamW, как один из наиболее быстрых и эффективных для обучения нейронных сетей.

Результаты обучения моделей можно видеть на рисунках 5.6 и 5.7.

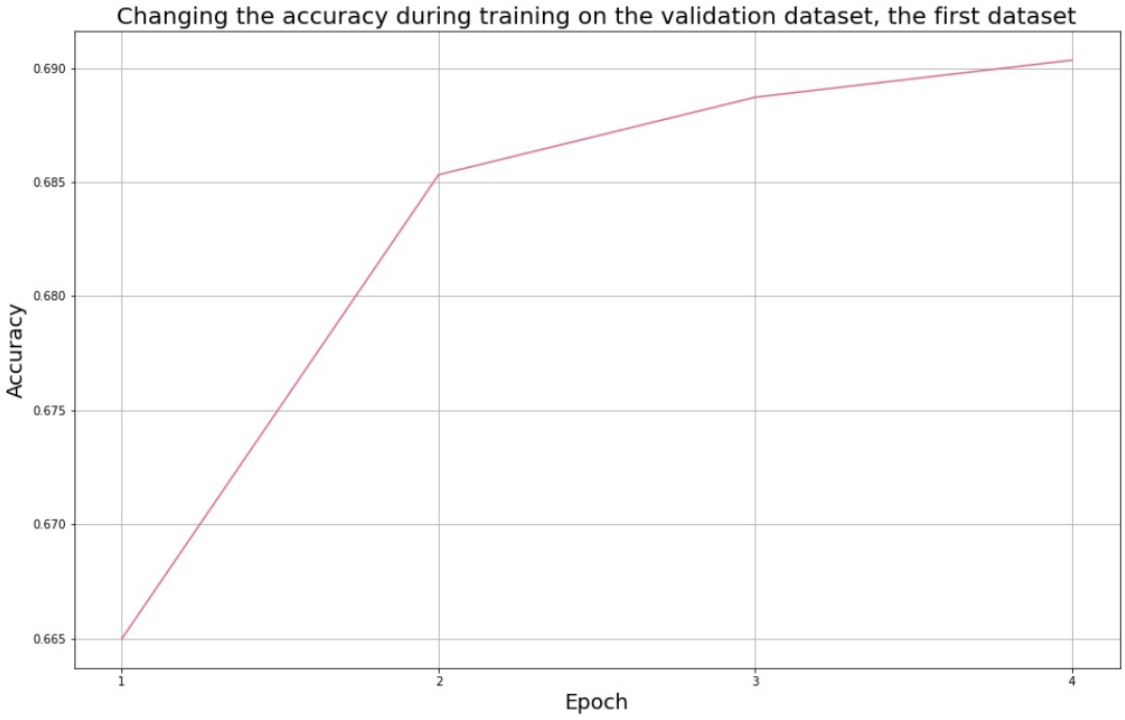

Рис. 5.6: Результаты обучения для первого набора данных

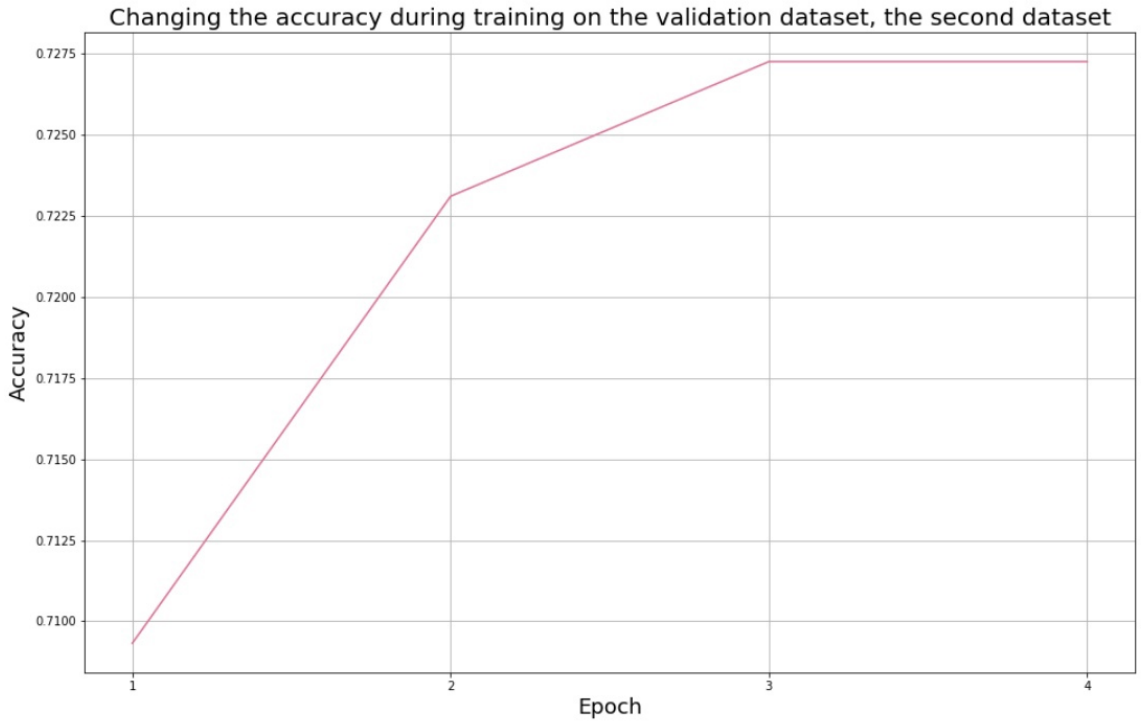

Рис. 5.7: Результаты обучения для второго набора данных

Как можно заметить, качество классификации получилось лучше на наборе данных «Amazon Fine Food Reviews». Возможно, это связано со спецификой отзывов на фильмы и еду. Кажется, что обычно отзывы на фильмы имеют более сложную структуру, поэтому полученные нами результаты вполне объяснимы.

Также может показаться, что качество полученных результатов не слишком хорошее. Но стоит вспомнить, что мы решали задачу многоклассовой классификации, а не бинарной. Если задуматься, то можно понять, что даже человеку сложно отличить отзыв на фильм с оценкой 4 и с оценкой 5, также и с низкими оценками. Для того, чтобы убедиться в этом на практике, был вручную размечен небольшой семпл данных (500 текстов рецензий) и проведено сравнение целевых метрик при такой разметке и при разметке обученной моделью. Результат вышел таковым, что accuracy для "ручной" разметки составило примерно 0.59, лучший результат модели GPT-2 же равен 0.75. Таким образом мы получили языковую модель, способную размечать тексты рецензий лучше, чем человек.

Результаты метрик при использовании микро и макро-усреднения пред-

ставлены в таблице 5.1.

Таблица 5.1: Полученные значения метрик для обоих наборов данных на валидационных выборках

	Micro – average				Macro – average			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Dataset 1	0.8892	0.7230	0.7230	0.7230	0.8892	0.5558	0.4777	0.5138
Dataset 2	0.8949	0.7372	0.7372	0.7372	0.8949	0.5550	0.4760	0.5124

Можно заметить, что в случае с микро-усреднением результаты получились лучше у модели, обученной на втором наборе данных, а в случае с макро-усреднением наоборт у модели, обученной первом на наборе данных. Что означает, что вторая модель более чувствительна к размеру классов, что в нашем случае необходимо из-за сильной несбалансированности классов.

5.3.2 Суммаризация

Следующим шагом после предобработки наборов данных стало непосредственно обучение модели GPT-2 для решения задачи суммаризации. На вход модели данные подавались следующим образом: текст рецензии + <SEP> (специальный токен для разделения) + саммари рецензии. Что в целом является стандартным видом подачи данных моделе в рамках задачи пересказа.

Задача суммаризации текста может рассматриваться как условная языковая модель, что значит мы генерируем вывод по заданному входу, то есть выход модели зависит от заданного условия – входных предложений. Обычная языковая модель выводит только вероятность определенного предложения (что используется для генерации новых предложений), тогда как условная выбирает наиболее вероятный вывод с учетом уже имеющегося входного предложения.

Архитектура модели разделена на 2 части – декодер и энкодер. В нашем случае входное предложение с помощью энкодера представляется в виде вектора и подаётся на вход декодера, где уже и возникают некоторые проблемы. Для конкретного входа может быть большое количество разных выходов и

неясно, какой предпочтительнее для модели. Для того, чтобы решить, какое предложение лучше выбрать в качестве выхода используется алгоритм beam search. Для выполнения алгоритма изначально задается параметр beam width = k (является обучаемым гиперпараметром для конкретной модели), который отвечает за количество вариантов слов на каждом шаге алгоритма. Сам же алгоритм работает следующим образом:

1) На первом шаге выбираются k наиболее вероятных слова.

2) Затем для каждого из выбранных k слов также выбираются 3 наиболее вероятных следующих слова.

3) Алгоритм продолжается до тех пор, пока либо не будет достигнута максимальная длина предложения, либо закончится пул возможных слов.

В работе был использован beam search с $k = 3$ (параметр был подобран методом перебора гиперпараметров). Также в модели использовался ядерное сэмплирование (nucleus sampling) Было протестировано несколько разных комбинаций параметров и выбраны следующие: $top\_k = 10$, $top\_p = 0{,}5$, температура = 0,8. При ядерном сэмплировании top_k отвечает за количество токенов с наибольшей вероятностью, которые будет рассматривать наша модель, добавляя в пул, после чего применяется beam search. top_p отвечает за общую сумму вероятностей токенов в пуле, предполагая, что вероятность должна быть $\geq top\_p$. А температура используется для управления случайностью прогнозов путем масштабирования логитов перед применением softmax.

Итоговая модель была дообучена на всех данных на 1 эпохе. Примеры результатов представлены далее. В целом, качество сгенерированных пересказов вышло далеко не идеальным и полученные тексты в случае набора данных «Amazon Review Data (2018). Movies and TV» можно разделить на несколько видов.

1) Приемлемые пересказы, соответствующие содержанию исходного текста и неплохо его отражающие (5.8 и 5.9).

2) Странного вида пересказы, зачастую с повторяющимися словами (5.10).

3) Пересказы, полностью цитирующие одно или несколько предложений исходного текста (5.11).

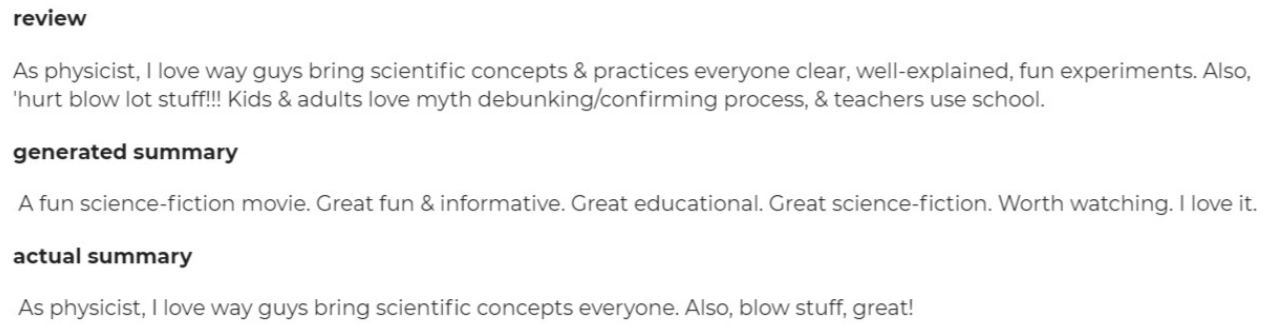

Рис. 5.8: Результат суммаризации для набора данных «Amazon Review Data (2018). Movies and TV». Пример 1.

Рис. 5.9: Результат суммаризации для набора данных «Amazon Review Data (2018). Movies and TV». Пример 2.

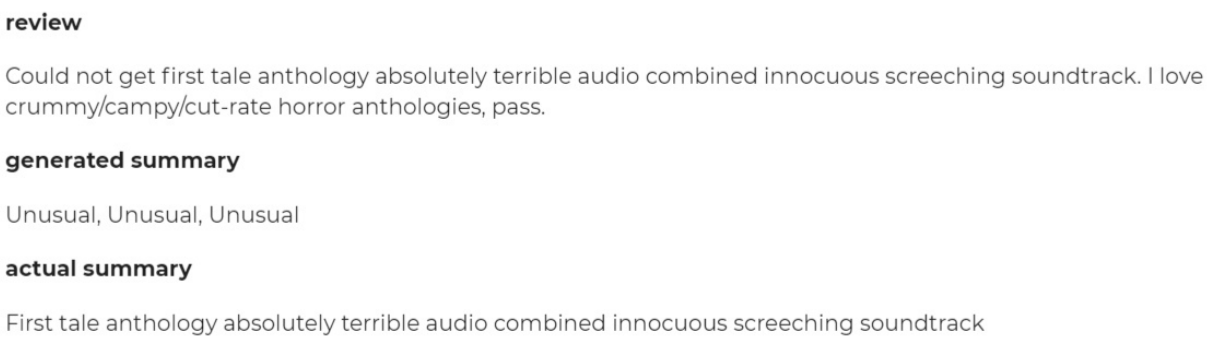

Рис. 5.10: Результат суммаризации для набора данных «Amazon Review Data (2018). Movies and TV». Пример 4.

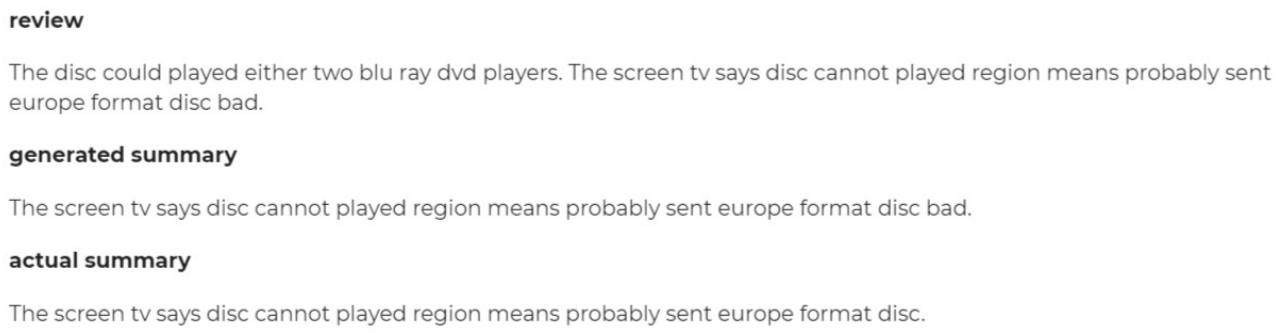

Рис. 5.11: Результат суммаризации для набора данных «Amazon Review Data (2018). Movies and TV». Пример 5.

Большая часть пересказов, полученных при помощи модели, обученной на первом наборе данных имеет слабую корреляцию с таргетом, то есть изначальными краткими содержаниями текста. Что в целом объяснимо, так как изначально в данных были не пересказы исходных текстов рецензий, а краткие отзывы, написанные людьми. Ясно, что они могут как совпадать по содержанию и тогда такой пример будет являться полезным для работы модели суммаризации, так и иметь абсолютно разное содержание.

Говоря о наборе данных «Amazon Fine Food Reviews», с ним дела обстоят немного иначе. В целом все краткие содержания, сгенерированные моделью, получились разумными. Но так как в случае этого набора данных целевой переменной у нас являлись названия рецензий, то выжимки также получились очень коротки и зачастую несущими в себе довольно общую информацию (5.12, 5.13).

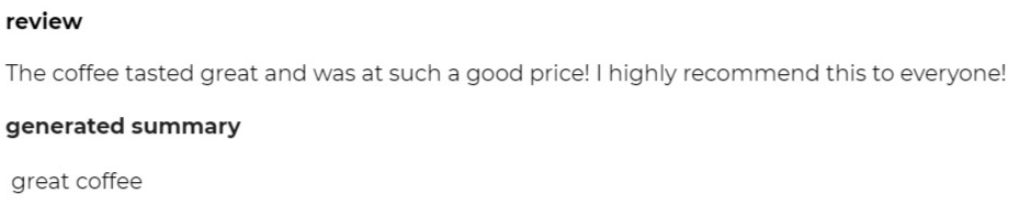

Рис. 5.12: Результат суммаризации для набора данных «Amazon Fine Food Reviews». Пример 1.

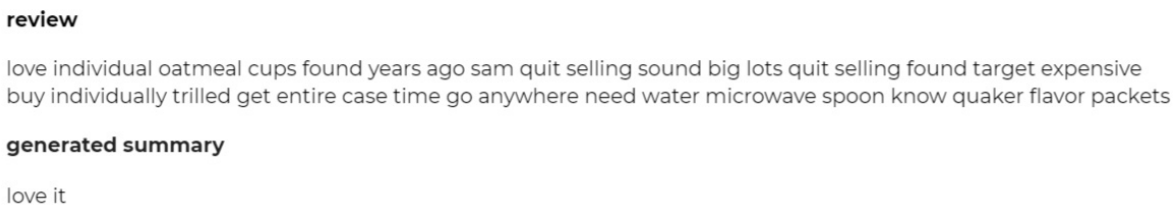

Рис. 5.13: Результат суммаризации для набора данных «Amazon Fine Food Reviews». Пример 2.

6 Заключение

В рамках данной курсовой работы было проведено исследование использования модели GPT-2 для решения задач классификации и суммаризации текстов рецензий. Рассмотрено два набора данных, каждый из которых использовался для решения поставленной задачи.

Исходя из полученных результов для многоклассовой классификации можно сделать вывод, что применяемая модель хорошо справляется с задачей даже при условии сильной несбалансированности классов. Также на практике было проведено сравнение с разметкой текстов рецензий человеком, в результате которого выяснилось, что GPT-2 справляется с данной задачей на порядок лучше.

По итогам решения задачи суммаризации можно прийти к выводу, что для её решения на высоком уровне необходимо иметь качественный датасет, содержащий не только отзывы, но и правильным образом написанные/сгенерированные пересказы. Иначе модель плохо подстраивается под целевую переменную, несмотря на то, что все-таки генерирует осознанные пересказы и выжимки текстов.

Таким образом, ряд проведенных экспериментов с GPT-2 показал, что данная модель имеет потенциал для решения задач классификации и суммаризации текстов и может применяться на практике при дополнительной её доработке.

Дальнейший возможный план работы:

1) Обучить большую версию модели (GPT-2 Large).

2) Попробовать совместить классификацию и суммаризацию в рамках одной

модели, тем самым решая задачу multitask learning.

3) С помощью различного рода эвристик попытаться улучшить модели.

7 Приложения

1. Материалы лекций и практические задания. Симагин Д.А.

2. Репозиторий github с решениями практических заданий, соревнований на Kaggle и кодом проекта.

3. Набор данных «Amazon Review Data (2018). Movies and TV»

4. Набор данных «Amazon Fine Food Reviews»

Список источников

[1] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 2015.

[2] Jason Wang, Luis Perez. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. arXiv preprint arXiv: 1712.04621, 2017.

[3] Sergey Ioffe, Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv preprint arXiv: 1502.03167, 2015.

[4] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, Martin Riedmiller. Striving for Simplicity: the All Convolutional Net. ICRL, 2015.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. arXiv preprint arXiv: 1301.3781, 2013.

[6] Jeffrey Pennington, Richard Socher, Christopher D. Manning. GloVe: Global Vectors for Word Representation. Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1532–1543, 2014.

[7] Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov. Enriching Word Vectors with Subword Information. arXiv preprint arXiv: 1607.04606, 2017.

[8] Klaus Greff, Rupesh K. Srivastava, Jan Koutnik, Bas R. Steunebrink, Jurgen Schmidhuber. LSTM: A Search Space Odyssey. arXiv preprint arXiv: 1503.04069, 2017.

[9] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, Luke Zettlemoyer. Deep contextualized word representations. arXiv preprint arXiv: 1802.05365, 2018.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin. Attention Is All You Need. arXiv preprint arXiv: 1706.03762, 2017.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv preprint arXiv: 1810.04805, 2019.

[12] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever. Improving Language Understanding by Generative Pre-Training. Unpublished preprint, 2018.

[13] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. Language Models are Unsupervised Multitask Learners. 2019.

[14] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger,

Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, Dario Amodei. Language Models are Few-Shot Learners. arXiv preprint arXiv: 2005.14165, 2020.

[15] Yang Liu, Mirella Lapata. Text Summarization with Pretrained Encoders. arXiv preprint arXiv: 1908.08345, 2019.

[16] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer. BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv preprint arXiv: 1910.13461, 2019.

[17] Tian Shi, Yaser Keneshloo, Naren Ramakrishnan, Chandan K. Reddy. Neural Abstractive Text Summarization with Sequence-to-Sequence Models. arXiv preprint arXiv: 1812.02303, 2020.

[18] Bowen Tan, Virapat Kieuvongngam, Yiming Niu. Automatic Text Summarization of COVID-19 Medical Research Articles using BERT and GPT-2. arXiv preprint arXiv: 2006.0199, 2020.

[19] David Currie. Text Summarization with Amazon Reviews. Unpublished, 2017.

[20] Kishore Papineni, Salim Roukos, Todd Ward, Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. 40th Annual Meeting of the Association for Computational Linguistics (ACL), pages 311-318, 2002.

[21] Chin-Yew Lin. ROUGE: A Package for Automatic Evaluation of Summaries. Unpublished preprint, 2004.

[22] Jincy B. Chrystal1 and Stephy Joseph, MULTI-LABEL CLASSIFICATION OF PRODUCT REVIEWS USING STRUCTURED SVM. IJAIA, Vol. 6, No. 3, 2015.

[23] Chi Sun, Xipeng Qiu, Yige Xu, Xuanjing Huang. How to Fine-Tune BERT for Text Classification? arXiv preprint arXiv: 1905.05583, 2020.

[24] Chloe Reams. GPT2 Sentiment Analysis. Unpublished, 2019.

[25] H Xu, B Liu, L Shu, PS Yu/ BERT post-training for review reading comprehension and aspect-based sentiment analysis. arXiv preprint arXiv: 1904.02232, 2019.