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

Факультет компьютерных наук
Основная образовательная программа
Прикладная математика и информатика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Исследовательский проект на тему "Определение оценочных и фактических суждений в новостных текстах"

Выполнил студент группы 192, 4 курса,
Лоптев Сергей Евгеньевич

Руководитель КР:
старший преподаватель Артёмова Екатерина Леонидовна

Москва 2023

Содержание

Аннотация

С расширением обсуждаемых тем, освещаемых в социальных сетях, читателям становится все труднее извлекать фактическую информацию из новостных статей, содержащих большое количество мнений. Эта задача также широко известна как задача обнаружения субъективности — отличить субъективные предложения от беспристрастных, или объективных. В этой статье представлено создание инструмента для выявления субъективности в новостных статьях с нуля. Во-первых, мы собираем современный набор данных новостных статей, аннотированных для субъективности работниками краудсорсинга. Затем мы настраиваем модель BERT, достигая точности, близкой к лучшим достигнутым результатам, в задаче обнаружения субъективности. Мы также используем продвинутые методы дообучения, такие как политика одного цикла и обучение с несколькими задачами. Наконец, мы создаем удобный инструмент для командной строки для автоматического аннотирования новостных статей с применением дообученной модели. Польза от этой работы выходит за рамки работы с новостями, поскольку точное распознавание мнений и фактов имеет важное значение для таких областей, как поиск информации и анализ тональности.

Ключевые слова — обработка естественного языка, обнаружение субъективности, краудсорсинг, сбор аннотированных данных, обработка новостей

With the expansion of topics covered on social media, it is becoming increasingly difficult for readers to extract factual information from news articles containing a large amount of opinions. This task is also commonly known as the subjectivity detection task — distinguishing subjective sentences from unbiased or objective sentences. This article presents the creation of a tool for detecting subjectivity in news articles from scratch. First, we collect a modern dataset of news articles annotated for subjectivity by crowdsourcing workers. We then tune the BERT model to achieve close-to-best-achieved accuracy in the subjectivity detection problem. We also use advanced learning methods such as the one cycle policy and multi-task learning. Finally, we are building a handy command-line tool for automatic annotation of news articles using a pre-trained model. The benefits of this work go beyond news analysis, as accurate recognition of opinions and facts is essential for areas such as information retrieval and sentiment analysis.

Keywords — natural language processing, subjectivity detection, crowdsourcing, annotated data collection, news processing

1 Введение

В данной статье рассматривается задача отличия фактов от мнений в новостных статьях. Эта задача также известна как задача обнаружения субъективности. Это важная подзадача анализа тональности, поскольку модели анализа тональности работают лучше, когда работают с корпусами, содержащими исключительно оценочные суждения.

В настоящее время многие действующие информационные агентства, в том числе BBC, The Guardian и другие, ежедневно публикуют тонны материалов. Количество горячо обсуждаемых тем сегодня также растет: среди недавних — экологические проблемы, COVID-19, гонка поисковых систем между Microsoft и Google и другие. С таким количеством обсуждаемых новостей обычному читателю становится трудно извлечь факты, чтобы иметь непредвзятый взгляд на происходящие события.

В этой статье рассматривается создание инструмента для различения фактических и субъективных предложений в новостных статьях. Инструмент позволяет пользователю загружать текст новостной статьи и аннотировать каждое предложение меткой факта или мнения.

Мы достигаем поставленной цели пошагово. Сначала мы создаём набор предложений из современных новостных статей. Этот набор данных аннотируется людьми на субъективность. Wiebe, J. et al. создали аналогичный набор данных под названием MPQA в [1]. Их исследование является основным аналогом нашей работы по созданию современного набора новостных данных. Аннотации предложений для этого набора данных были созданы в 1999 году в [2] и сейчас являются устаревшими. Во-вторых, мы дообучаем модель BERT для обнаружения субъективности. Pang, B. et al. рассмотрели различные методы дообучения модели BERT в [3]. Мы переиспользуем и проверяем результаты их исследования — они достигли точности около 84%, что верифицируется в данной статье. В-третьих, мы создаём новый инструмент, который по ссылке на новостную статью извлекает ее текст, размечает

его на предмет субъективности и выдаёт размеченную статью. В настоящее время у данного инструмента нет аналогов; ближайший похожий инструмент — Grammarly, который может помочь обнаружить слишком оценочные предложения и адаптировать текст под настроенные критерии.

Структура работы выглядит следующим образом. Раздел 2 охватывает научные статьи, связанные с нашим исследованием, то есть работы, охватывающие создание как наборов данных, так и моделей. В разделе 3 мы описываем нашу методологию для сбора и человеческой разметки набора данных. В разделе 4 мы подробно описываем дообучение модели BERT. В разделе 5 описывается создание CLI-инструмента для использования дообученной модели. В разделе 6 подробно описываются полученные результаты. Раздел 7 содержит выводы статьи.

2 Анализ литературы

Достаточное количество исследований было проведено в области обнаружения субъективности. Существует несколько различных наборов данных, предназначенных для этой задачи в нескольких областях. В [4] был представлен набор данных SUBJ. Он состоит из 5000 субъективных и 5000 объективных предложений, взятых с веб-сайтов IMDb и RottenTomatoes, аннотированных с помощью алгоритма. В [5] создается набор данных, содержащий утверждения из Википедии с тегами POV (point of view). Утверждения с тегом POV — это утверждения, которые, как сообщается в статье, нарушают принцип NPOV (neutral point of view) Википедии. Эти утверждения затем аннотируются людьми, и в конечном итоге набор данных состоит из 1843 предвзятых утверждений и 3109 нейтральных. В [1] был представлен набор данных MPQA (Multi-Perspective Question Answering). Он содержит предложения из 535 испанских новостных статей, переведенных на английский язык и размеченных авторами на субъективность. Всего в наборе данных около 9700 предложений, и около 55% из них субъективны. Все эти наборы данных имеют отношение к

нашей задаче, но только MPQA удовлетворяет нашей области, то есть новостным статьям. Однако набор данных MPQA был создан около 20 лет назад. Мы считаем, что за 20 лет стиль новостных текстов в информационных агентств изменился; кроме того, были введены новые темы, такие как COVID-19. Было создано несколько новых агентств, а некоторые старые обанкротились. По всем этим причинам мы решили создать более современный набор данных, который в целом будет похож на MPQA, но будет состоять из современных данных.

Большой объём работ был также проделан на тему аннотации текстов. В процессе разработки набора данных MPQA авторами Wiebe, J., et al. были выпущены статьи [6], [7] и [8]. Во многом они использовались как вдохновители для нашей инструкции по разметке данных, в том числе, некоторые примеры были взяты из этих статей. При создании набора данных SUBJ, в статье [4], за изначальные данные были взяты отзывы с сайтов RottenTomatoes (отмечены как полностью субъективные) и IMDb (отмечены как полностью объективные), и взяты близости предложений из неразмеченного набора данных с размеченными. Затем был использован алгоритм min-cut-max-flow, использующий потоки, для разметки оставшейся части набора данных. В статье [5], самой новой из тех, что представляют новый набор данных, был использован краудсорсинг (сервис Amazon Mechanical Turk) — аннотаторам предлагались предложения и достаточно простая инструкция; необходимо было выбрать один из трёх вариантов ответов. В настоящее время краудсорсинг стал наиболее популярной техникой разметки наборов данных, что подтверждается тем, что статья [5] первая из нами рассмотренных стала использовать эту технику.

Помимо создания наборов данных, большой объем работы был посвящен обучению моделей для выявления субъективности. В [9] показан временной прогресс таких моделей. Первые методы использовали примитивные функции, например, обнаружение ключевых слов. Затем исследователи перешли к онтологическим моделям, то есть к определению набора онтологий, которые определяют отношения между разными классами слов и проецируют их в

векторное пространство. Следующими были статистические методы, похожие на классическое машинное обучение — модели, обученные на аннотированном наборе данных. Одним из примеров такой модели является Passive-Aggressive Classifier, упомянутый в [10]. Авторы этого исследования утверждают, что их метод достиг около 85% F1-score на кросс-валидации. Это показывает, что ранние подходы к задаче были уже достаточно успешными. Следующими моделями были так называемые LDM (Latent Dirichlet Models), которые использовали частоту слов для вычисления апостериорного распределения. Более новые методы используют глубинное обучение для обнаружения субъективности. Например, существуют сверточные модели (CNN) для обнаружения субъективности, где CNN служат основной моделью, а затем ответ извлекается с помощью RNN или других методов. Один из таких методов описан в [11] — они используют байесовскую сетевую машину экстремального обучения (BNELM) поверх CNN для достижения точности 89% на TASS 2015 — наборе данных, содержащем твиты на испанском языке для обнаружения субъективности. Наконец, новейшими моделями обнаружения субъективности являются модели, основанные на архитектуре трансформеров, такие как BERT. Одним из таких исследований является [3], где авторы достигают точности от 84% до 95% на разных наборах данных. Они также пробуют различные методы дообучения для повышения качества и обнаруживают, что такие методы, как политика одного цикла и обучение с несколькими задачами являются наиболее полезными. Мы собираемся повторить этот эксперимент в нашей работе.

Как упоминалось ранее, в настоящее время нет аналогов нашей работы именно для обнаружения субъективности. Существуют средства для написания новостей, такие как Headline Analyzer, Ahrefs и Grammarly, но нет инструментов для автоматического анализа новостей.

3 Сбор и разметка данных

3.1 Выбор исходного набора данных

Перед разметкой необходимо подобрать неразмеченный набор данных. Нашими главными критериями для подбора такого набора данных были:

- Достаточный размер: одно из главных требований к нашему размеченному набору данных было наличие достаточного количества субъективных и объективных предложений для эффективного дообучения модели BERT, поэтому необходимо было, чтобы неразмеченный набор данных был достаточно большим.

- Наличие вариативности в новостных источниках: наш инструмент в итоге должен получиться достаточно универсальным, так что с его помощью можно было бы размечать как, например, новости про спорт, так и новости про экологию.

- Возможность расширения: по возможности, способ сбора набора данных должен быть достаточно простым, чтобы можно было собрать самые новые статьи.

- Простота использования: наша работа нацелена более на аннотацию, чем на сбор данных, поэтому нужен набор данных, который будет легко использовать без дополнительной обработки.

Среди вариантов наборов данных, предложенных на сайте HuggingFace[1], были рассмотрены следующие:

- news_commentary[2] — набор данных с переводом большого количества статей между различными языками. Не был выбран, так как данные изначально были собраны не для классификации, и понадобилась бы

[1] https://huggingface.co/

[2] https://huggingface.co/datasets/news_commentary

дополнительная обработка, чтобы вывести из него нужные статьи на английском языке.

- multi_news[3] — набор данных для суммаризации, в котором для суммаризации даются статьи из разных источников. Не был выбран по той же причине: этот набор данных был собран для другой задачи, и была необходима предобработка, чтобы достать статьи в нужном нам формате.

- argilla[4] — набор данных, содержащий статьи, изначально собранные для классификации. Содержит около 20000 статей, что подходит нам, но не был выбран, так как нет возможности собрать его заново из самых новых статей.

- cc_news[5] — набор данных, состоящий из статей, собранных с помощью утилиты news-please[6], которую легко использовать, и потенциально можно было бы использовать также для итогового инструмента командной строки. Этот набор данных прост в использовании, легко расширяется, а также содержит сотни тысяч статей, собранных из различных новостных порталов, поэтому он был выбран в качестве неразмеченного набора данных для нашей задачи.

3.2 Подготовка исходного набора данных к аннотации

Выбранный набор данных содержал несколько сотен тысяч статей, что было слишком много для аннотации. Необходимо было отобрать столько самых подходящих статей, чтобы суммарно в них было около 20000 предложений. Также необходимо было разделить эти статьи на предложения, чтобы подготовить данные к разметке. Для этого были проделаны следующие шаги:

[3]https://huggingface.co/datasets/multi_news
[4]https://huggingface.co/datasets/argilla/news-summary
[5]https://huggingface.co/datasets/cc_news
[6]https://github.com/fhamborg/news-please

- Из набора данных были убраны все статьи, содержащие менее пяти предложений — мы признали эти статьи выбросами из общего распределения.

- Статьи были дедуплицированы — набор данных мог содержать одинаковые статьи из одного и того же источника, но разных веб-сайтов, например, reuters.co.uk и reuters.com.

- В оставшемся наборе данных были оставлены только источники, содержащие от 100 статей. Это было сделано для удаления слишком редких источников — вероятных выбросов из общего распределения.

- Были выбраны самые новые статьи, так, чтобы суммарное число предложений было около 20000. В итоге было оставлено 1024 статьи из июля 2018 года, содержащие в сумме 19953 предложения. Мы посчитали, что июль 2018 года — это достаточно новые статьи, и можно просто взять их, и таким образом, сбор новейших данных и разработка инструмента, использующего их, остаётся для будущей работы.

3.3 Написание инструкции

Значительное количество времени было выделено на написание качественной инструкции. Для этого был предпринят следующий алгоритм действий. Изначально была составлена инструкция на основе [8], адаптированная под нашу конкретную задачу. Пользователям предлагалось выбрать либо опцию "неприменимо предназначенную для предложений, не содержащих никакие утверждения и, следовательно, не подходящих для разметки, либо степень субъективности по дискретной шкале от 1 до 5, также известной как шкала Ликерта. Дальше происходили три итерации улучшения инструкции, состоящие в том, что автор и третье лицо размечали три заранее выбранных статьи и сравнивали результаты. После одной из итераций состоялась консультация с студентом Школы лингвистики НИУ ВШЭ, в ходе которой инструкция

была улучшена. После трёх итераций удалось получить корелляцию 0.62 и F1-score 0.41 между авторскими ответами и ответами третьего лица. Так как особенность задачи состоит в субъективности разметки, такие результаты были признаны достаточно хорошими, чтобы запускать разметку полного набора данных.

Получившаяся инструкция на английском языке доступна в Приложении А к данному документу.

3.4 Разметка с помощью краудсорсинга

Для более масштабной разметки был использован сервис Toloka AI[7]. В данном сервисе есть два важных понятия: проект и пул. Проект — это сущность, содержащая несколько пулов. На уровне проекта задаётся инструкция и некоторые правила контроля качества. Пул — это набор данных, который непосредственно размечают работники краудсорсинга. Бывают разные виды пулов, например, тренировочный пул создан для того, чтобы на примерах объяснить работникам инструкцию; этот пул размечается бесплатно. Экзаменационный пул необходим, чтобы отобрать работников с достаточным умением, и отсеять мошенников и роботов. Далее, существует общий вид пула, в котором работники уже непосредственно размечают рабочие данные. В нашем проекте были все эти три вида пулов. Для тренировочного пула было специально написано несколько примеров под каждую метку; также были написаны пояснения к этим примерам. Для экзаменационного пула были использованы те размеченные автором данные, которые использовались при улучшении инструкции. Остальные данные были размечены с помощью общего пула. Параметр перекрытия (то есть, сколько аннотаторов должны разметить одно предложение) был выбран равным 3. В качестве правил контроля качества использовались ограничения по числу размеченных наборов задач на человека за день, ограничение по пропуску наборов задач подряд, а также отложенная

[7]https://toloka.ai/

приёмка (то есть, перед оплатой денег разметка должна была быть проверена вручную, и аннотации могли быть отклонены). Изначально было размечено всего 5 статей, чтобы удостовериться в правильности настройки. После того, как настройки были улучшены, а разметка этих пяти статей выглядела адекватно, были запущены в разметку остальные статьи.

3.5 Пост-обработка разметки

Перед агрегацией ответов аннотаторов метки были отображены в более удобное пространство для модели: метки “1” и “2” отображаются в метку “объективное”, метка “3” в метку “нейтральное”, метки “4” и “5” в метку “субъективное”, метка “неприменимо” в себя. Распределение меток доступно в таблице 6.1.

Затем для непосредственно агрегации был применён алгоритм WAWA[8]. Это итеративный алгоритм, который в начале выдаёт каждому работнику вес 1, потом вычисляет для каждого предложения голос взвешенного большинства и перераспределяет веса между работниками в зависимости от того, как близко они к этому большинству на каждом примере. После агрегации голосов аннотаторов для каждого работника был посчитан F1-score, и задачи, размеченные аннотаторами, получившими F1-score меньше 0.35, были переразмечены. Таким образом, нам удалось достичь достаточного уровня согласия между разными аннотаторами.

Далее, данные были поделены на тренировочную, валидационную и тестирующую выборки в пропорции $0.72 : 0.08 : 0.2$.

4 Обучение модели

В рамках данной статьи мы дообучили модель bert-base-cased[9] на размеченном наборе данных. Мы поставили две задачи в рамках этого раздела:

[8] https://toloka.ai/docs/crowd-kit/reference/crowdkit.aggregation.classification.wawa.Wawa/

[9] https://huggingface.co/bert-base-cased

получить достаточно хорошее качество на тестирующей выборке — то есть, сравнимое с качеством, полученным в статье [3] — точность около 0.84, и протестировать успешные эксперименты из статьи [3], а именно политику одного цикла и обучение с несколькими задачами.

4.1 Детали реализации и обучения

Обучение было реализовано на языке Python, с применением библиотеки PyTorch. В качестве оптимизатора использовался Adam. Базовая модель была взята с сайта HuggingFace, как и токенизатор. В качестве головы модели были взяты слой Dropout с параметром $p = 0.1$ и линейный слой. Для каждого эксперимента в рамках оптимизации гиперпараметров мы обучали финальную модель на 5 эпох и брали модель с эпохи, на которой получилась лучшая точность на валидационной выборке. В рамках каждого эксперимента менялся какой-то один параметр, остальные фиксировались. Реализация доступна на сервисе GitHub[10].

Обучение производилось на GPU NVidia A100 SXM4, GPU RAM 39.9GB, Total GPU TFlops 19.5. Машина была арендована на сервисе vast.ai.

4.2 Трансформация данных

Перед оптимизацией гиперпараметров было необходимо отобрать данные, которые мы бы подавали в модель. При выборе трансформации мы сначала обучили модель на нетрансформированных обучающих данных, вручную проанализировали её ошибки, и затем на основе этих ошибок попробовали трансформировать данные. В качестве экспериментов были рассмотрены следующие трансформации:

- Так как модель была сильно смещена в сторону объективных предложений и переобучалась на них, мы попробовали сбалансировать классы

[10]https://github.com/beastSL/hse-thesis/tree/main/model

в обучающей выборке — взяли случайную подвыборку объективных предложений, чтобы в итоге их было столько же, сколько субъективных.

- Так как модель почти никогда не предсказывала нейтральную метку, мы попробовали убрать из всех выборок примеры с нейтральной меткой. Данное решение в любом случае не повлияло бы на продуктовую составляющую: нам бы хотелось полностью размечать новостную статью на субъективность.

- Так как модель работала хуже из-за отсутствия контекста, мы попробовали добавить ко всем примерам предыдущее и следующее предложение из статьи.

Результаты данной оптимизации доступны в разделе 6.

4.3 Выбор модели

Также необходимо было выбрать модель, которую мы будем дообучать. Изначально мы хотели дообучать BERT, поэтому выбор был лишь между bert-base-cased и bert-base-uncased[11]. Вторая модель использовалась в статье [3], но наши данные содержат буквы в верхнем регистре. Результаты тестирования данных моделей приведены в разделе 6.

4.4 Изначальная оптимизация гиперпараметров

Перед тем, как применять продвинутые техники оптимизации, было необходимо оптимизировать базовые гиперпараметры. Мы оптимизировали скорость обучения, максимальное число токенов, которое можно пропустить в модель и параметр β_2 оптимизатора Adam. Кроме этого, мы попробовали заменить модель bert-base-uncased на bert-base-cased[12]. Результаты данной оптимизации представлены в разделе 6.

[11] https://huggingface.co/bert-base-uncased
[12] https://huggingface.co/bert-base-cased

4.5 Политика одного цикла

В качестве одной из продвинутых техник дообучения модели BERT было взято расписание скорости обучения, называемое “политика одного цикла”, представленное в [12]. Согласно этому расписанию, скорость обучения за все запланированные эпохи проходит через две фазы: нагревание (меньшая часть обучения, при которой скорость обучения поднимается с $\mathrm{lr}_{\mathrm{init}}$ до $\mathrm{lr}_{\mathrm{max}}$) и охлаждение (большая часть обучения, при которой скорость обучения спускается обратно). Опционально также есть третья фаза: аннигиляция, при которой скорость обучения в конце обучения опускается дальше до $\mathrm{lr}_{\mathrm{min}}$. Такое расписание придумано с целью предотвратить спуск к локальному высокому минимуму в начале обучения, затем ускорить обучение в зоне с хорошим градиентом (где-то посередине обучения), и затем уменьшать скорость обучения, чтобы не выйти из хорошего локального минимума. Также есть опция изменять импульс скорости обучения: можно также задать $\mathrm{momentum}_{\mathrm{max}}$ и $\mathrm{momentum}_{\mathrm{min}}$, но в этом случае импульс будет изменяться в обратную сторону: от максимального к минимальному и обратно (фазы аннигиляции для импульса не предполагается). Изменение импульса и самой скорости обучения показано на графике 4.1.

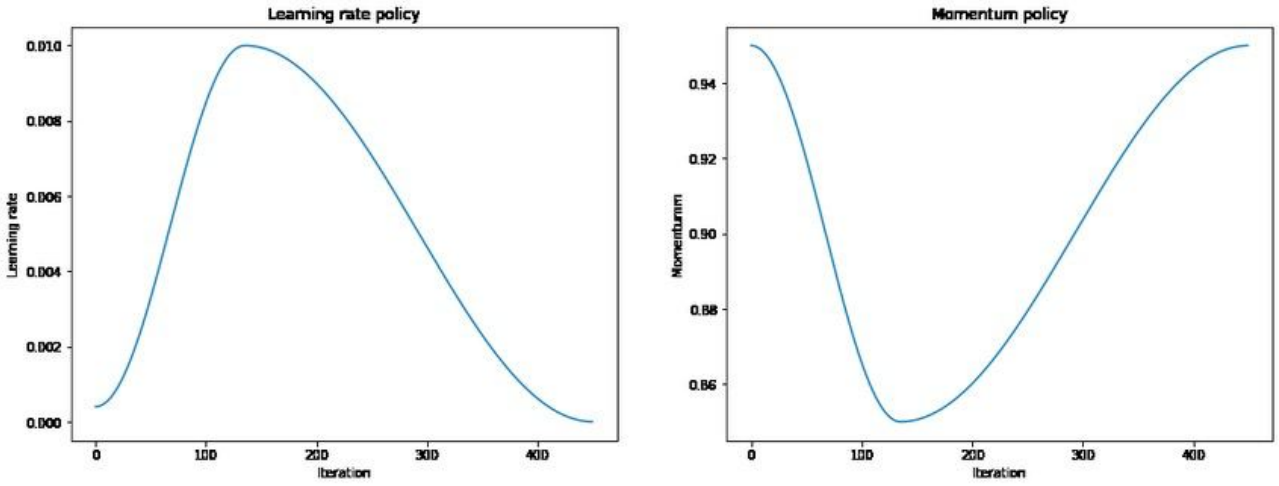

Рис. 4.1

Для выбора одного из наборов параметров, задающих скорость обучения при политике одного цикла, было применено “тестирование диапазона скорости

обучения", описанное в [13]. Его суть заключается в следующем. Необходимо найти минимальную скорость обучения, при которой ошибка модели начинает расходиться. Эта скорость берётся за $\mathrm{lr}_{\max}$. Затем, $\mathrm{lr}_{\mathrm{init}}$ задаётся как $\frac{\mathrm{lr}_{\max}}{10}$, и $\mathrm{lr}_{\min} = \frac{\mathrm{lr}_{\max}}{100}$. Кроме этого набора параметров скорости обучения мы также пробовали оптимизировать другие (детали доступны в разделе 6).

4.6 Обучение с несколькими задачами

В качестве второй продвинутой техники дообучения было принято дообучение модели на нескольких задачах. Эта техника впервые была представлена в статье [14]. Её суть заключается в том, что вместо того, чтобы дообучать модель и классификатор на одну задачу, мы будем обучаться на разные задачи, чередуя наборы данных из этих разных задач для оптимизации. Было показано, что данный подход успешно работает во многих приложениях, связанных с обработкой естественного языка и компьютерным зрением.

4.6.1 Наборы данных

В качестве задач мы взяли следующие наборы данных:

- Разумеется, мы взяли наш набор данных. Для удобства, здесь и далее мы будем обозначать его как NSDC (News Subjectivity Detection Corpus). На выходе будет ожидаться одна из трёх меток: неприменимо, объективное предложение, или субъективное предложение.

- Также мы взяли набор данных SUBJ из статьи [4], в котором содержится 10000 предложений из обзоров фильмов, размеченных алгоритмически на субъективность (по 5000 на каждый класс). Детали разметки данного набора данных приведены в секции 2. Соответственно, на выходе ожидается одна из двух меток: объективность/субъективность.

- В качестве ещё одного набора данных в задаче обнаружения субъективности мы взяли набор данных Wikipedia biased statements из статьи

[5]. Данный набор данных был собран из предложений в Википедии, которые по мнению некоторых пользователей были отмечены содержащими чью-то субъективную точку зрения. Этот набор данных содержит 3109 объективно сформулированных предложений и 1843 субъективно сформулированных предложений. Детали разметки данного набора данных приведены в секции 2. Таким образом, это задача классификации отдельных предложений, ожидающая одну из двух меток на выходе.

- Также один из взятых наборов данных — IMDb, представленный в статье [15]. Он состоит из 50000 предложений из отзывов, взятых с сайта IMDb, размеченных на задачу анализа настроения (то есть, на задачу определения позитивности/негативности предложения). Данный набор данных также решает задачу классификации отдельных предложений, и ожидает одну из двух меток на выходе.

- Мы также взяли два набора данных для задачи классификации текстов. Один из них — набор данных SNLI (Stanford Natural Language Inference), представленный в статье [16]. Он содержит 570152 пар предложений, вручную размеченных на соотношение второго предложения с первым, то есть возможны три метки: логическое следствие, нейтральность и противоречие.

- Второй набор данных для задачи классификации текстов — QNLI (Stanford Question Answering dataset), представленный в статье [17]. Он содержит 115669 пар вопрос-параграф, размеченный на задачу бинарной классификации: содержит ли параграф ответ на данный вопрос. В данном наборе данных вопросы были написаны аннотаторами вручную, а параграфы взяты из Википедии.

- Наконец, для задачи регрессии мы взяли набор данных STS-B (Semantic Textual Similarity Benchmark), представленный в статье [18]. Он содержит 8628 пар предложений из заголовков новостей, подписей к видео

и картинкам, и других источников. Для каждой пары предложений необходимо оценить их близость по шкале от 0 до 5.

Сводная информация по всем наборам данных представлена в таблице 4.1.

Набор данных	**Число примеров**	**Количество меток**	**Задача**
NSDC	19953	4	Обнаружение субъективности, классификация предложений
SUBJ	10000	2	Обнаружение субъективности, классификация предложений
Wikipedia biased statements	4952	2	Обнаружение субъективности, классификация предложений
IMDb	50000	2	Классификация предложений
QNLI	115669	2	Классификация текстов
SNLI	570152	3	Классификация текстов
STS-B	8628	1	Регрессия

Таблица 4.1: Сводная информация о данных

4.6.2 Детали реализации

Мы реализовали обучение с несколькими задачами следующим образом. Для каждого набора данных был введён отдельный линейный классификатор, при этом предобученная модель BERT оставалась общей для всех задач. Одна эпоха обучения состояла из поочерёдного обучения модели и классификатора на соответствующей задаче. При этом для оптимизации классификации использовался PyTorch модуль `CrossEntropyLoss`, а для оптимизации регрессии — `MSELoss`. Визуально это представлено на рисунке 4.2. После мультизадачного обучения модели на несколько эпох мы сохраняли модель и пробовали дообучить её на целевом наборе данных ещё на 5 эпох. В итоге сохранялась версия модели, дающая лучший результат на валидационной выборке.

Для оптимизации гиперпараметров мы пробовали перебирать множество задач, на которых обучалась модель и количество эпох, в течение которых

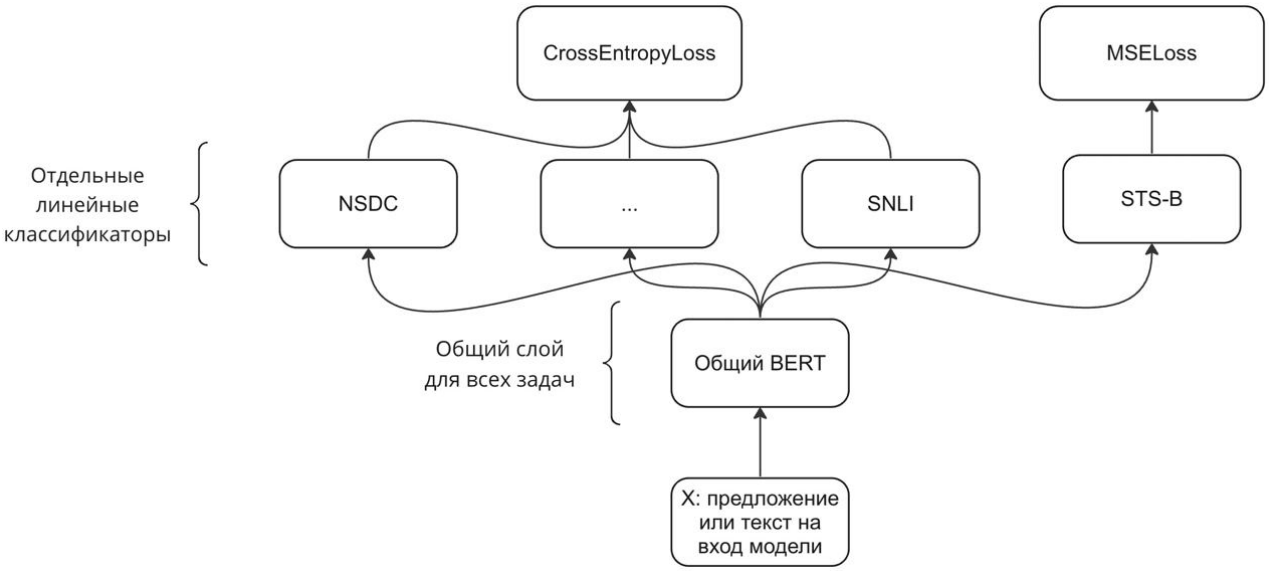

Рис. 4.2: Архитектура для обучения на несколько задач

модель обучалась на несколько задач. Результаты доступны в секции 6.

5 Создание CLI-инструмента

Для использования обученной модели мы создали CLI-инструмент. В рамках этой работы был реализован алгоритм, который по URL новостной статьи запускает локальный HTTP-сервер, на котором находится HTML-страница с размеченным на субъективность текстом новостной статьи. Данный алгоритм состоит из следующих шагов:

1 На вход алгоритму подаётся URL новостной статьи и директория, из которой будет запускаться HTTP-сервер.

2 С помощью библиотеки news-please мы скачиваем новостную статью и сохраняем её заголовок, текст, разделённый на абзацы, и главную картинку.

3 С помощью обученной модели мы размечаем весь текст на субъективность. Таким образом, каждому предложению сопоставляется либо метка “неприменимо к разметке”, либо метка “объективное”, либо метка “субъективное”.

4 Мы создаём текст HTML-документа с сохранёнными заголовком статьи, главной картинкой и размеченным текстом. Объективные предложения выделяются зелёным фоном, а субъективные — красным. Мы сохраняем этот HTML-файл в указанную директорию, из которой пбудет запускаться сервер.

5 С помощью команды `python3 -m http.server` мы запускаем локальный HTTP-сервер, содержащий сгенерированую веб-страницу.

Также мы добавили опцию указывать вместо URL веб-страницы файл с текстом статьи. Такая альтернатива может быть полезна в некоторых применениях, например, если скачать текст статьи с помощью библиотеки news-please не представляется возможным.

Далее, мы опросили трёх человек об их опыте использования данного CLI-инструмента. Результаты доступны в секции 5.

6 Результаты

6.1 Сбор данных

Мы собрали современный набор данных, размеченных на задачу обнаружения субъективности, с помощью разметки работниками краудсорсинга. Мы получили достаточно большой уровень согласия аннотаторов — F1-score каждого аннотатора против агрегированных меток составляет не менее 0.35. Итоговое распределение меток доступно в таблице 6.1.

Результаты смещены в сторону объективных предложений, что ожидаемо, так как новости обычно пишутся в объективном стиле. Тем не менее, нам удалось собрать достаточно большое количество субъективных предложений, что достаточно для обучения модели, и поэтому сбор данных можно считать успешным.

Метка	Количество предложений с данной меткой
Неприменимо	1265
Объективное	14255
Нейтральное	516
Субъективное	3917

Таблица 6.1

6.2 Обучение модели

В данном разделе мы опишем результаты оптимизации гиперпараметров для обучения модели, а также вручную проанализируем типичные ошибки, которые она допускает.

6.2.1 Базовые модели

В качестве базовых моделей, от которых можно было бы отталкиваться при оценке модели, мы использовали классификатор, всегда выдающий самый популярный вариант, и логистическую регрессию, обученную на признаках TF-IDF. Результаты применения этих моделей доступны в таблице 6.2.

Модель	Лучшая точность на валидационной выборке	Лучшая F1-мера на валидационной выборке
Константный классификатор	0.7406	0.2837
TF-IDF + логистическая регрессия	0.774	0.4839

Таблица 6.2: Результаты выбора данных.

6.2.2 Трансформации данных

Для начала, мы протестировали различные трансформации данных для более эффективного дообучения модели. Мотивация, стоящая за выбором

таких трансфромаций, изложена в секции 4. Результаты данной оптимизации доступны в таблице 6.3. Так как отсутствие нейтральной метки заметно улучшило F1-меру, мы решили оставить эти данные. В следующих секциях все эксперименты проводились именно на таких данных.

Данные	**Лучшая точность на валидационной выборке**	**Лучшая F1-мера на валидационной выборке**
Данные по умолчанию	0.8228	0.5733
Сбалансированная обучающая выборка	0.7676	0.5737
Данные без нейтральной метки	**0.8255**	**0.7535**
Данные с контекстом	0.713	0.4071

Таблица 6.3: Результаты выбора данных.

6.2.3 Выбор модели

Также необходимо было выбрать модель, в рамках чего мы протестировали модели bert-base-uncased и bert-base-cased с гиперпараметрами по умолчанию. Результаты тестирования доступны в таблице 6.4. Как видно из результатов, мы решили выбрать модель bert-base-cased, так как выбор идёт в первую очередь по F1-мере.

Модель	**Лучшая точность на валидационной выборке**	**Лучшая F1-мера на валидационной выборке**
bert-base-uncased	**0.8255**	0.7535
bert-base-cased	0.8213	**0.7594**

Таблица 6.4: Результаты выбора модели. Параметры, использованные при выборе: максимальное число токенов (max_len — 512, скорость обучения (lr) — 2e-5, параметр оптимизатора β_2 — 0.99).

6.2.4 Изначальная оптимизация гиперпараметров

В качестве изначальной оптимизации гиперпараметров мы оптимизировали скорость обучения, максимальное число токенов, которое можно пропустить в модель и параметр β_2 оптимизатора Adam. В качестве сетки для скорости обучения были взяты значения $5 \cdot 10^{-6}$, 10^{-5}, $2 \cdot 10^{-5}$ (значение по умолчанию), $5 \cdot 10^{-5}$ и 10^{-4}. В качестве сетки для максимального числа токенов мы взяли 256 и 512 (значение по умолчанию). В качестве сетки для параметра β_2 мы взяли 0.98, 0.99 (значение по умолчанию) и 0.999. Результаты данной оптимизации доступны в таблице 6.5.

Эксперимент	**Лучшая точность на валидационной выборке**	**Лучшая F1-мера на валидационной выборке**
Гиперпараметры по умолчанию	0.8213	0.7594
max_len = 256	0.8213	0.7594
lr = 10^{-4}	0.8115	0.737
lr = $5 \cdot 10^{-5}$	0.8053	0.7644
lr = 10^{-5}	**0.8255**	0.767
lr = $5 \cdot 10^{-6}$	0.8282	0.7639
$\beta_2 = 0.999$	0.8282	0.7771
$\beta_2 = 0.98$	**0.8255**	**0.7798**

Таблица 6.5: Результаты изначальной оптимизации гиперпараметров. Значения по умолчанию такие: максимальное число токенов (max_len — 512, скорость обучения (lr) — 2e-5, параметр оптимизатора β_2 — 0.99).

Прокомментируем данные результаты. Мы смогли улучшить F1-меру на валидационной выборке на 0.02, присвоив параметру β_2 значение 0.98. Это не очень сильное улучшение, но от оптимизации таких параметров и не ожидался сильный прирост качества. При этом комбинирование разных экспериментов хороших результатов не дало.

6.2.5 Политика одного цикла

Здесь и далее будем считать, что за модель по умолчанию была принята модель, обученная на данных без нейтральной метки с гиперпараметрами оптимизатора по умолчанию, кроме β_2 (со значением 0.98). Мы реализовали политику одного цикла и попробовали оптимизировать её гиперпараметры. Мы оптимизировали следующие гиперпараметры: параметры скоростей обучения ($\mathrm{lr}_{\mathrm{min}}$, $\mathrm{lr}_{\mathrm{max}}$, $\mathrm{lr}_{\mathrm{init}}$), наборы импульсов ($\mathrm{momentum}_{\mathrm{max}}$ и $\mathrm{momentum}_{\mathrm{min}}$), пробовали дообучать с фазой аннигиляции.

В библиотеке PyTorch $\mathrm{lr}_{\mathrm{init}}$ и $\mathrm{lr}_{\mathrm{min}}$ задаются через $\mathrm{lr}_{\mathrm{max}}$ с помощью делителей div_factor и final_div_factor с помощью следующих уравнений:

$$\begin{cases} \mathrm{lr}_{\mathrm{init}} = \dfrac{\mathrm{lr}_{\mathrm{max}}}{\mathrm{div\_factor}} \\ \mathrm{lr}_{\mathrm{min}} = \dfrac{\mathrm{lr}_{\mathrm{max}}}{\mathrm{final\_div\_factor}} \end{cases}.$$

По умолчанию $\mathrm{lr}_{\mathrm{max}} = 2 \cdot 10^{-4}$, $\mathrm{div\_factor} = 25$, $\mathrm{div\_factor} = 10^4$, причём фаза аннигиляции отключена. С такими делителями мы перебирали $\mathrm{lr}_{\mathrm{max}}$ по сетке $[5 \cdot 10^{-6},\ 10^{-5},\ 2 \cdot 10^{-5},\ 5 \cdot 10^{-5},\ 10^{-4}]$. Отдельно мы пробовали $\mathrm{div\_factor} = 10$ и включить фазу аннигиляции со стандартным final_div_factor. Также мы пробовали изменить параметры импульса. По умолчанию в PyTorch $\mathrm{momentum}_{\mathrm{min}} = 0.85$, $\mathrm{momentum}_{\mathrm{max}} = 0.95$, мы попробовали значения из статьи — 0.8 и 0.9 соответственно. Также мы попробовали скомбинировать параметры $\mathrm{div\_factor} = 10$ и $\mathrm{final\_div\_factor} = 100$, включив фазу аннигиляции. Результаты доступны в таблице 6.6.

Комментируя эти результаты, можно подытожить, что политика одного цикла не внесла положительного эффекта в обучение модели.

6.2.6 Обучение с несколькими задачами

Вторая продвинутая техника, которую мы реализовали — обучение с несколькими задачами. Здесь гиперпараметра всего два — множество задач,

Эксперимент	**Лучшая точность на валидационной выборке**	**Лучшая F1-мера на валидационной выборке**
Без политики одного цикла	0.8255	**0.7798**
Гиперпараметры по умолчанию	0.8157	0.7605
$\text{lr}_{\max} = 5e-5$	**0.8268**	0.772
$\text{lr}_{\max} = 2e-5$	**0.8268**	0.7637
$\text{lr}_{\max} = 1e-5$	0.8227	0.7542
$\text{lr}_{\max} = 5e-6$	0.8157	0.7508
div_factor = 10	0.8178	0.7515
$\text{momentum}_{\min} = 0.8$ $\text{momentum}_{\max} = 0.9$	0.822	0.7527
С фазой аннигиляции	0.8206	0.7667
div_factor = 10 final_div_factor = 10 С фазой аннигиляции	0.8178	0.7564

Таблица 6.6: Результаты оптимизации гиперпараметров политики одного цикла. Значения по умолчанию такие: фаза аннигиляции отключена, $\text{lr}_{\max} = 10^{-4}$, div_factor = 25, final_div_factor = 10^4, $\text{momentum}_{\min} = 0.85$, $\text{momentum}_{\max} = 0.95$.

на которых мы обучаем модель перед финальным дообучением на целевом наборе данных, и количество эпох мультизадачного обучения. Мы пробовали дообучать модель на целевом наборе данных (NSDC) после любого количества эпох, после которых ошибка модели на валидационном наборе данных продолжала падать. Множества задач были взяты следующие:

- Множество задач определения субъективности — NSDC, SUBJ, Wikipedia biased statements. Здесь пробовали дообучать после одной и двух мультизадачных эпох.
- Множество задач классификации одного предложения — NSDC, SUBJ, Wikipedia biased statements, IMDb. Здесь пробовали обучать после одной мультизадачной эпохи.
- Всё множество наборов данных, описанных в секции 4. Здесь пробовали обучать после одной и двух мультизадачных эпох.

При выборе числа эпох для такого “предобучения” мы руководствовались ошибкой модели на целевом наборе данных после каждого прохода через все наборы данных. Если ошибка модели продолжала идти вниз после какой-то эпохи, то позже мы пробовали дообучить модель, начиная с сохранения модели, сделанного в конце этой эпохи. Результаты оптимизации гиперпараметров доступны в таблице 6.7.

Эксперимент	**Лучшая точность на валидационной выборке**	**Лучшая F1-мера на валидационной выборке**
Без обучения с несколькими задачами	**0.8255**	**0.7798**
I, после 1 эпохи	0.8248	0.7471
I, после 2 эпох	0.8317	0.7563
II, после 1 эпохи	0.8303	0.7558
III, после 1 эпохи	0.8317	0.7601
III, после 2 эпох	0.8303	0.7572

Таблица 6.7: Результаты оптимизации гиперпараметров обучения с несколькими задачами. I, II и III — множества наборов данных, в порядке, описанном выше.

Комментируя эти результаты, можно подытожить, что обучение с несколькими задачами не внесло положительного эффекта в обучение модели.

6.2.7 Обсуждение и анализ ошибок модели

Лучший полученный результат — F1-мера 0.7798 и точность 0.8255. Этот результат сопоставим с результатом, полученным в статье [3] на наборе данных Wikipedia biased statements, равным 0.84. Также можно упомянуть, что данная F1-мера намного лучше базовых моделей — логистическая регрессия, обученная поверх признаков TF-IDF даёт F1-меру всего около 0.48. Поэтому в целом можно считать, что обучение модели прошло успешно.

Тем не менее, хотелось бы провести качественный анализ ошибок модели, чтобы понять направления дальнейшего улучшения. Мы просмотрели 100

примеров из тестирующей выборки, на которых модель отвечала неправильно. В этом разделе мы обсудим основные свойства этих ошибок и возможные направления для улучшения модели.

Для начала, для каждого из 100 примеров мы проверили, действительно ли модель не права. Оказалось, что модель действительно ошибается лишь в 51% случаев. Это неудивительно, так как задача обнаружения субъективности сама по себе крайне субъективна. Во многих случаях сложно точно количественно оценить, насколько предложение субъективно, особенно не имея контекста.

Из тех предложений, в которых модель ошибалась, особенно заметны следующие тенденции:

- Модель не воспринимает контекст. На вход модели всегда подаётся лишь одно предложение, но, как мы упоминали раньше, контекст достаточно важен при выявлении субъективности. Например, в случае прямой речи на несколько предложений, модель не будет знать о том, что предложения, находящиеся посередине этой прямой речи, принадлежат ей, и поэтому будет трактовать их неверно. Данная проблема отражена в 12 из 51 ошибочного примера. Вероятно, модель можно улучшить, подав на вход вдобавок к текущему предыдущее и следующее предложения, но данный эксперимент не дал нам прироста к качеству.

- Модель хуже аннотаторов понимает, когда выражается мнение третьего лица. В инструкции есть параграф про то, что предложения вида “Он сказал, что ...” считаются объективными; в целом, предложения такого вида следует судить по словам, которые указывают на прямую речь, такие как “сказать” или “осудить”. Данная проблема отражена в 8 из 51 ошибочного примера. Вероятно, эти ошибки можно было бы исправить, классифицировав предложения на те, которые говорят о мнениях человека в первом и третьем лице.

- Остальные ошибки модели не поддаются какой-то общей тенденции, но можно сказать, что почти все они вызваны неверным пониманием

семантики предложения.

В таблице 6.8 приведены примеры, иллюстрирующие вышеописанные тенденции.

Предложение	**Аннотаторская метка**	**Метка модели**	**Пояснение**
"At the end it is the American consumer who will pay the price for Mr Trump's policy.	1	3	В оригинальном тексте указателем на эту прямую речь было слово said, поэтому правильная метка — объективное предложение. Но этот контекст не подаётся в модель, поэтому она ошибается.
"We have spoken a lot about togetherness," Kane said, "and we've got a great bond off the pitch."	1	3	Слово said в этом предложении маркирует прямую речь, и так как оно нейтральное, предложение сформулировано объективно. Модель периодически не справляется с пониманием этого правила.
Perhaps Amazon is earning all its government contracts on a level playing field.	3	1	В некоторых случаях модель попросту не справляется семантически понять, что предложение субъективное.

Таблица 6.8: Примеры, иллюстрирующие некоторые тенденции в ошибках модели и аннотаторов. Напомним, что метки значат следующее: 0 — неприменимо к разметке, 1 — объективное предложение, 3 — субъективное предложение.

6.3 CLI-инструмент

Для использования проделанной нами работы мы создали CLI-инструмент, работающий с новостными статьями по URL. Необходимо упомянуть, что так как текст новостей собирался с помощью библиотеки news-please, для некоторых новостных порталов (например, BBC) данный инструмент не работает. Для обработки таких сервисов был написан режим работы инструмента, при котором на вход подаётся файл с самим текстом статьи. Тем не менее, многие известные издания, такие, как The Guardian, Daily Mail, Reuters и многие другие поддерживаются данной библиотекой.

После создания CLI-инструмента мы опросили трёх человек об их опыте использования данного инструмента. Их отзывы доступны в таблице 6.9. Комментируя эти отзывы, можно подытожить, что инструмент действительно полезен при прочтении новостных статей, а негативная часть этих отзывов совпадает с тенденциями, которые мы заметили при анализе ошибок модели.

Отзыв
Так, ну, я почитал, вчитываясь в саму новость, цвета помогают не забывать, что именно ты читаешь — сообщение о факте или чьё-то чужое мнение про факт. У меня сложилось ощущение, что иногда оно похожие цитаты разным цветом красит, но вероятно, показалось. Идея крутая.
На глаз видно, что он не совсем точно разделяет субъективные и объективные штуки, как будто некоторым предложениям не хватает контекста от предыдущих и от этого разделение странное получается. При этом текст, помеченный субъективным, читать обычно интереснее — вероятно, потому что в них содержатся мнения.
При прочтении статьи с выделенными субъективными предложениями у меня сложилось ощущение, что этим предложениям доверять нельзя. Особенно если это цитаты или вставки из диалогов. Это помогает не верить на 100% информации из этих предложений, а наоборот, задумываться, насколько вообще эта мысль верна. Помогает сложить свое мнение на этот счет.

Таблица 6.9: Отзывы об опыте использования CLI-инструмента.

7 Заключение

В этой статье описана работа по созданию нового инструмента для распознавания мнений и фактов в новостных статьях. Был собран набор данных, необходимый для эффективного дообучения модели, проведено само дообучение, а также создан CLI-инструмент для применения данной модели. Мы уверены, что этот проект значительно повлияет на анализ новостей, и инструмент будет достаточно удобным для публичного распространения. Также мы считаем, что данная работа, в частности, создание такого набора данных, окажет положительное влияние на развитие сферы обнаружения субъективности.

Список литературы (или источников)

1. J. Wiebe and E. Riloff, "Creating subjective and objective sentence classifiers from unannotated text," in *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, 2005, pp. 486–497.

2. J. Wiebe, R. Bruce, and T. O'Hara, "Development and use of a gold standard data set for subjectivity classifications," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 1999)*, 1999.

3. H. Huo and M. Iwaihara, "Utilizing bert pretrained models with various fine-tune methods for subjectivity detection," in *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*. Springer, 2020, pp. 270–284.

4. B. Pang and L. Lee, "A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts," in *Proceedings of ACL 2004*, 2004.

5. C. Hube and B. Fetahu, "Neural based statement classification for biased language." in *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. ACM, 2019, pp. 195–203.

6. J. Wiebe, "Instructions for annotating opinions in newspaper articles," University of Pittsburgh, Tech. Rep., 2002.

7. J. Wiebe, T. Wilson, and C. Cardie, "Annotating expressions of opinions and emotions in language," *Language resources and evaluation*, vol. 39, pp. 165–210, 2005.

8. T. Wilson and J. Wiebe, "Annotating attributions and private states," in *CorpusAnno '05: Proceedings of the Workshop on Frontiers in Corpus Annotations II*, 2005.

9. I. Chaturvedi, E. Cambria, R. Welsch, and F. Herrera, "Distinguishing between facts and opinions for sentiment analysis: survey and challenges," *Information Fusion*, vol. 44, pp. 65–77, 2018.

10. A. Stepinski and V. Mittal, "A fact/opinion classifier for news articles," in *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. ACM, 2007, pp. 807–808.

11. I. Chaturvedi, E. Ragusa, P. Gastaldo, R. Zunino, and E. Cambria, "Bayesian network based extreme learning machine for subjectivity detection," *Journal of The Franklin Institute*, vol. 355, pp. 1780–1797, 2018.

12. L. N. Smith, "A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay." US Naval Research Laboratory, Tech. Rep., 2018.

13. ——, "Cyclical learning rates for training neural networks," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, Ed., 2017.

14. R. Caruana, "Multitask learning: A knowledge-based source of inductive bias," *Machine Learning*, vol. 28, 1997.

15. A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Human Language Technology*, 2011, pp. 142–150.

16. S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. EMNLP, 2015, pp. 632–642.

17. P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000 + questions for machine comprehension of text," in *EMNLP*, 2016.

18. D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "Semeval2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop Semantic Evaluation*, 2017, pp. 1–14.

A Инструкция для аннотаторской разметки

Мы прикладываем неотредактированную инструкцию на английском языке, которая была непосредственно доступна аннотаторам при разметке данных.

Annotation instructions

This document describes the annotation instructions for subjectivity detection in news articles. First, we will describe what should be treated as subjectivity and objectivity and how to spot it. Then, we will fully describe the annotation task.

Subjectivity and objectivity

First, we would like to note there are no formal definitions of subjectivity and objectivity. In many cases, you will have to appeal to your intuition and your reaction after reading a sentence. However, we will try to advise on how to notice subjectivity and help build up the needed intuition.

Types of subjectivity

Subjectivity is an expression that represents opinions, beliefs, thoughts, feelings, emotions, goals, evaluations, and/or judgments (we'll use the expression "private states"to generalize these concepts). One key sign of it is that it is not open to objective observation or verification. Subjective expressions are the ones that cannot be refuted or confirmed.

The main types of subjectivity are:

- direct mentions of the private state. This is the easiest-to-spot sign of subjectivity, including words that directly indicate private states. Examples:
  - The U.S. fears a spill-over.
  - I really enjoyed the book I read last night.

These sentences are highly subjective because the words "fear"directly indicate the emotion of fear, and the word "enjoyed"directly indicates the emotion of joy.

- expressive subjective elements. These are words or expressions in the text that implicitly indicate subjectivity. Here are some examples:
  - The report is full of absurdities.
  - The sunset painted the sky with a fiery orange hue.

  We think these sentences are subjective because the expressions "full of absurdities"and "fiery"represent the writer's opinions or emotions.

Note that this is not an exhaustive list of subjectivity types. In some cases, you'll need to apply your intuition and appeal to the reaction a sentence gives you.

Subjectivity when describing private states of a third party

A common case is sentences describing the private states (opinions, emotions, etc.) of a third party. One good example of such a description is direct or indirect speech (see the first two examples). Here are some examples of such sentences:

- Sargeant O'Leary said the incident took place at 2:00 pm.
- Defence officials accused Beijing of using President Tsai's US visit as an "excuse to conduct military exercises".
- These people remember the horrors of World War II.

When annotating such sentences, you should not base your judgment on the private states of the third party themselves (e.g. "the incident took place at 2:00pm"). Instead, you should base your judgments on how these descriptions are presented in the sentence, and whether a certain tone is given to the private states. For example, words like "say "know and "want"are neutral, while words

like "fear"and "accuse"give an intonation to the private states. Note that this way we are incentivizing you to spot specifically the writer's subjectivity.

We think sentences 1 and 3 are objective, because the words "said"and "remember"sound neutral, and the second sentence is subjective, because the word "accused"does indicate a tone of accusation.

Objective sentences

The sentences that do not contain any subjectivity above and that present statements are considered objective. Note that these statements are not necessarily correct. Some examples:

- The Earth is flat.
- The Dow Jones Industrial Average closed at 34,035.99 points on Monday.

Other important advice

The other important things we need to mention before describing the exact task:

- There are no fixed rules about how particular words should be annotated. The instructions describe the annotations of specific examples but do not state that specific words should always be annotated a certain way.
- Sentences should be interpreted with respect to the contexts in which they appear. You should not take sentences out of context and think about what they could mean but rather should judge them as they are being used in that particular sentence and document.
- It is impossible to cover all types of sentences in this instruction. For example, there could be sentences containing both objective and subjective elements. The subjective elements can also play a minor role in the sentence. You should base your judgment on your inner reaction and intuition after reading a sentence.

Task

You will be consequently given sentences from a newspaper. Every sentence will be surrounded by several adjacent sentences to provide context, but the current sentence you're labelling will be highlighted.

Before labelling the main pool, you will need to pass training and an exam. Note that in order to get paid you will need to get 35% correct responses on the exam. Also note that if your responses are on average too far from the majority vote on the main pool, your responses will be looked through and can be rejected.

Please note that some sentences might contain explicit language since the papers were scraped from the Internet.

Your task is to assign each sentence a subjectivity score. The score will be measured on a discrete scale from 1 to 5. You will also be given the option to assign a "Not applicable"label. Here are the explanations of the scale:

- The "Not applicable"label is used when a sentence does not contain any statements, and therefore it is impossible to say if it is subjective or objective. Some cases for the "Not Applicable"label are incomplete sentences, questions and sentences fully consisting of noise. (see below for examples)

- Score 1 should be assigned when you are confident that the sentence is objective.

- Score 2 should be assigned if you are unconfident but suspect that the sentence is objective.

- Score 3 should be assigned if a sentence presents a statement but it is difficult to say whether the sentence is objective or subjective. Note that this option corresponds to the case when neither the instruction nor the intuition can help to decide if the sentence is subjective or objective, even though the annotation applies.

- Score 4 should be assigned if you are unconfident but suspect that the sentence is subjective.

- Score 5 should be assigned when you are confident that the sentence is subjective.