ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ ФГАОУ ВО
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

UDC 004.62, 004.8

Отчет об исследовательском проекте

на тему Активное обучение на ошибках в экстремальной мультиклассовой классификации коротких текстов

Выполнил:

студент группы БПИ202 _____ Бугриенко Данил Артурович
подпись

Дата 05.05.2022

Принял:

Дранга Даниил Игоревич, Data Science Community Lead, Райффайзенбанк

Дата 05.05.2022 _____ _____
Оценка (по 10бальной шкале) подпись

Москва 2022

Содержание

1 Реферат

Современное машинное обучение с ростом популярности нейросетевых подходов требует все больше и больше качественно размеченных данных для обучения. В то же время, сферы его применения постоянно расширяются, охватывая также те, где разметка данных крайне ресурсоемка, что затрудняет достижение необходимого качества алгоритмов. Разрешением этой проблемы занимается активное обучение, призванное сокращать количество объектов для разметки датасета, за счет оценки ожидаемого увеличения качества модели от добавления семпла в обучающую выборку.
В данной работе будут предложены и провалидированы обобщения классических алгоритмов активного обучения, позволяющие улучшить качество классификации на задачах с большим датасетом и множеством (более 100) классов за счет выборочной разметки ограниченного числа элементов. Подход предполагает упор на обобщенность (работу независимо от природы данных датасета и сбалансированности классов в нем) на основе одной или нескольких различных по природе моделей машинного обучения и пространственной структуры векторных представлений данных.

2 Введение

Значительная часть современных задач в сфере машинного обучения требует больших объемов размеченных данных. В зависимости от сложности предметной области разметка каждого нового образца датасета может выполняться в автоматическом режиме при помощи математических вычислений / стандартных методик программирования за сравнительно низкую стоимость и малое количество времени, в других случаях может потребоваться вмешательство человека, привлечения значительных временных и денежных ресурсов. В отдельных примерах от асессора могут требоваться специфические знания в доменной области, что может увеличить стоимость и время разметки на порядки. В таких случаях для минимизации требуемых ресурсов полезно еще до начала разметки знать, какие элементы изначальной выборки больше всего улучшат качество модели. Сфера машинного обучения, которая решает подобные задачи называется active learning [16].
Задача классификации с большим количеством классов представляется дополнительную сложность, потому что существует множество схожих между собой объектов, принадлежащих к разным классам, и необходимо выстроить алгоритм активного обучения таким образом, чтобы в равной степени оптимизировать точность предсказаний на каждом из представленных классов.
Данное исследование направлено на перенос и тестирование методов классического активного машинного обучения на задачу экстремальной многоклассовой классификации коротких текстов.

3 Аналитический обзор литературы

3.1 Извлечение признаков из текстов

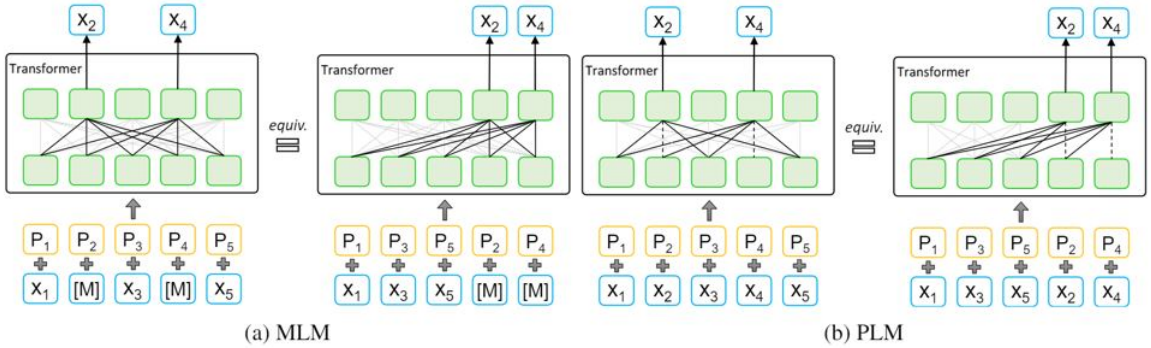
Методы машинного обучение не могут работать с текстами напрямую. Поэтому необходимо научиться представлять исходные тексты векторами признаков.

Поскольку мы не обладаем достаточно большой выборкой, чтобы строить наши признаки на уровне целых текстов или предложений (почти все их представители уникальны, что не дает извлечь из них много информации), поэтому будем извлекать признаки из уровня слов и/или сочетаний букв.
Прежде всего, в исходных данных не содержится никакой информации о границах отдельных слов, поэтому мы вынуждены устанавливать их самостоятельно. Эту задачу способны решить средства для токенизации текстов, например, токенизатор из библиотеки NLTK [8], работающий на основе регулярных выражений. Однако существуют более продвинутые способы решения данной задачи, например униграмные модели.
Такая модель получает на вход корпус текста и строит по нему частоты сегментов слов. После этого она принимает на вход текст и возвращает его наиболее вероятное разбиение на сегменты, которые встретились ей в обучающей выборке [17], где вероятность разбиения

$$P(x_1, ..., x_n) = \prod_{i=1}^{n} p(x_i)$$

x_i - одна из наиболее вероятных последовательностей букв из исходном корпусе.
$p(x_i)$ - вероятность встретить такую последовательность в исходном корпусе.
Одна на этом этапе мы будем иметь множество одинаковых по смыслу слов, отличающихся некоторыми морфемами, поэтому необходимо привести все слова к исходному виду. Для этого слова лемматизируются - приводятся к начальной форме путем анализа их морфологического состава.
Теперь, когда у нас есть достаточно чистый токенизированный датасет с начальными формами слов, мы можем начать его векторизировать.
Самым простым вариантом в этом случае будет представление текста вектором, отражающим количество вхождений каждого уникального слова корпуса в этот текст - bag of words. Однако такой подход сохраняет минимальное количество информации о тексте: никак не учитывается контекст появления слов и их индивидуальная ценность, а также он вынуждает на хранить разреженные матрицы размера количество слов в словаре $*$ количество текстов. Мы можем приблизительно оценить ценность индивидуального слова для классификации подсчетом частоты его встречаемости внутри каждого документа умноженной на частоту встречаемости между документами. Данный способ получил название TF-IDF. Однако он по прежнему не учитываем совстречаемость слов в тексте и требует матрицы больших размеров для хранения. Мы можем учесть контекст слов, посчитав вероятность встречи слов вместе в контексте определенной из фиксированного количества слов. Такой подход решит проблему учета контекста и позволит получить гораздо более информативные преставления текста, но потребует значительных вычислительных и временных ресурсов для посчета, и также много памяти. В современном же NLP для представления слов применяются эмбеддинги - векторные представления, основанные на совстречаемости слов в определенных контекстах.
Высокоуровнево, получение векторных представлений слов представляет собой обучение модели, параметрами которой являются вектора, а целью - предсказание наиболее вероятных слов к заданному в контексте фиксированного размера. Например, word2vec [1] минимизирует расстояние между словами, которые встречаются в схожих контекстах, максимизируя расстояние с остальными.
Также существует FastText [2], который отличается работой не на уровне слов,

а на уровне букв и их сочетаний. FastText способен получить векторное представление для слова, не встречавшегося в обучающем корпусе, а также требует меньшие по размеру датасеты для обучения, так как он извлекает информацию не только из взаимного расположения слов.
И последним из рассматриваемых является GloVe [3], который в отличие от W2V учитывает также и частоту встречаемости слов в тренировочному корпусе.
Следующим идет вопрос, как вычислять векторные представления текстов. Одними из самых простых способов являются взвешенное суммирование отдельных векторов или суммирование с обучаемыми весами. Также могут быть применены сверточные или рекурентные нейронные сети для учета контекста, а также более сложные нейросетевые методы связанные с вниманием [27].
Примером подобного state-of-the-art метода является MPNet [32]. Высокая предсказательная способность модели достигается за счет совмещения двух подходов к моделированию естественного языка MPL (masked language modelling), заключающийся в обучении модели предсказывать исключенные из текста слова по их контексту, и подхода PML (permuted language modelling), который предполагает восстановление следующего токена по предыдущему контексту при условии, что все слова в предложении расположены в случайном порядке (оба подхода проиллюстрированы ниже). MLM ограничен неспособностью выучить зависимости



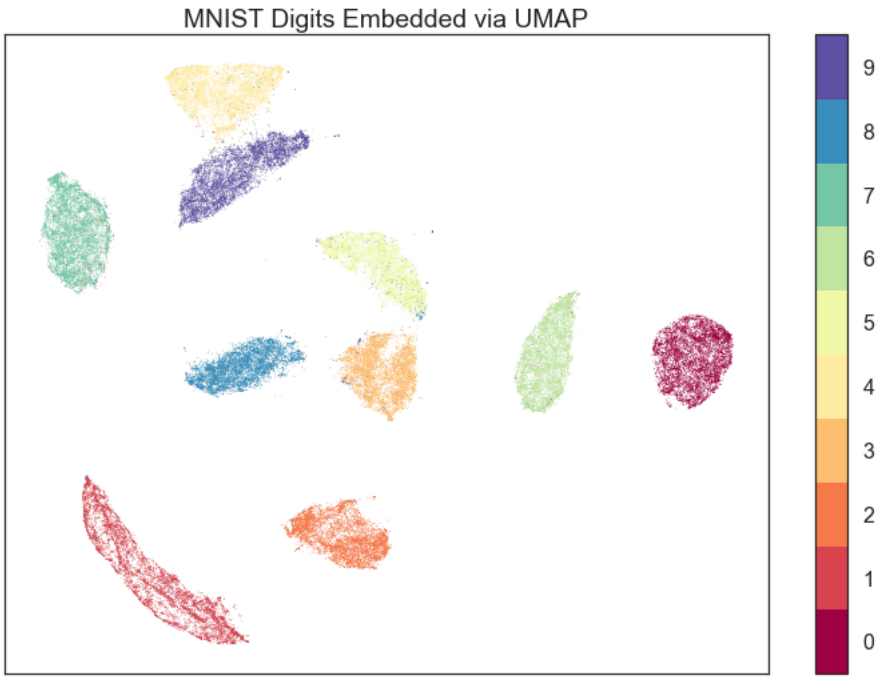
Схемы работы MLM и PLM [32]

между предсказанными словами, а также плохим пониманием сложных синтаксические конструкции. PLM в свое время во время обучения никогда не знает порядок слов во всем предложении, что вызывает проблемы при дальнейшем дообучении на конкретные прикладные задачи. MPNet решает данные проблемы за счет представления концепции предсказанных и непредсказанных токенов и управления направленностью внимания при их помощи.

3.2 Визуализация

Удостовериться в правдивости информации, содержащейся в эмбеддингах слов и построенными на них представлениями позволяют алгоритмы, проецирующие многомерные признаковые пространства на 2-х или 3-х мерную плоскость. Одним из наиболее успешно зарекомендовавших себя алгоритмов является UMAP [23], который позволяет проецировать многомерное векторное представление элементов датасета с сохранением глобальной структуры и пространственных отношений между объектами. Для этого алгоритм строит многомерный граф, представляющий данные, а потом пытается построит граф меньшей размерности, максимально похожий на исходный. Для построения графа строится fuzzy simplicial

complex, который является взвешенным графом, где веса означают вероятность соединения точек между собой, вероятность уменьшается с увеличением расстояния между точками [24].



Цифры из датасета MNIST, спроецированные на плоскость при помощи UMAP

3.3 Методы активного обучения

Далее предстоит решить проблему определения наиболее информативных образцов исходного пула данных. Базовыми методами оценки информативности является применение least confidence, margin и entropy семплирований [16].
Least confidence: из всех предсказанных вероятностей класса выбирается максимальная и принимается за меру информативности объекта - чем она ниже, тем больше информативность.
Margin семплирование - частично разрешает недостаток первого метода - он учитывает информацию только о классе, вероятность которого максимальна. Выбирая две максимальные вероятности и считая абсолютную разницу между ними можно получить метрику информативности превосходящую просто семплирование по наименьшей уверенности. Чем она больше - тем более информативен объект.
Entropy семплирование идет дальше и учитывает уже вероятность каждого класса, считая энтропию [19] вектора вероятностей. Однако, ориентируясь на предсказания только одной модели, мы можем упустить часть информации, которую ее архитектура не способна извлечь. Для борьбы с этой проблемой было предложено семплирование на основе несогласия нескольких моделей с разными архитектурами - Query by committee [20]. Наиболее информативные объекты получают различающиеся предсказания от разных моделей с высокой степенью

уверенности. До этого момента мы опирались исключительно на предсказания моделей, но в случае, если большинство самых информативных точек будут относиться к одному классу и располагаться близко друг к другу? Их разметка повысит точность модели на этом конкретном классе, но скорее всего не сильно улучшит интегральное качество модели. Чтобы обеспечить разнообразие новых точек в работах [21, 22] применяются методы кластеризации для повышения разнообразия отбираемых для разметки объектов, также может быть применено семплирование топа по неуверенности моделей.
Также существует более вычислительно затратный метод Expected Error Reduction [28]. Его идея заключается в том, чтобы, не имея доступ к истинным меткам объектов, предсказать, насколько отдельные объекты неразмеченной выборки уменьшат значение функции потерь модели. Формула оценки ожидаемого уменьшения ошибки для задачи классификации:

$$\tilde{E}_{\hat{P}_{\mathcal{D}^*}} = \frac{1}{|X_{pool}|} \sum_{x \in p} \sum_{y \in \mathcal{Y}} \hat{P}_{\mathcal{D}^*}(y|x) \log(\hat{P}_{\mathcal{D}^*}(y|x))$$

$\tilde{E}_{\hat{P}_{\mathcal{D}^*}}$ - ожидаемое значение лосс-функции модели на неразмеченной выборке
X_{pool} - неразмеченная выборка
$\hat{P}_{\mathcal{D}^*}$ - модель, обученная на размеченном датасете с добавлением единственного (x, y) из неразмеченной выборки, где y принимает все значения всех известных классов.
Данная процедура повторяется с последовательным переобучением модели на каждом экземляре неразмеченной выборки. В качестве приближения истинного распределения классов выбирается наша свежеобученная модель. Однако такой подход не учитывает, что модель может одновременно на десятках или даже сотнях тысяч объектов, из-за чего добавление к тренировочной выборке одного нового элемента (как предполагает подход оценки error reduction) практически не изменить градиент при обучении, а учитывания масштабы датасетов, многочисленные переобучения моделей могут оказаться невозможными даже при применении всевозможных оптимизаций (предварительная фильтрация кандидатов, приближение EER семплированием, алгоритмы частичного обучения моделей).
До этого момента мы опирались исключительно на предсказания моделей, но такие подходы обладают определенными уязвимостями: что если конкретная модель плохо подходит для датасета или предсказания информативности на нем в частности? Чтобы обеспечить разнообразие размечаемых объектов и повысить общее качество подхода в работах [21, 22] применяются методы кластеризации для повышения разнообразия отбираемых для разметки объектов, основывающееся на предположении, что общая информативность каждого кластера будет падать по мере набора новых точек из такового. Еще одним из способов улучшения уже перечисленных алгоритмов является учитывание плотности распределения объектов неразмеченной выборки. [21]. Однако статьи на тему пространственной структуры никак не учитывают и не исследуют тот факт, что в пространствах высокой размерности меры расстояния, а соотвественно плотности могут оказываться не очень релевантными и информативными, поэтому подобные методы могут оказаться неэффективными на больших зашумленных датасетах.

3.4 Exploration & exploitation tradeoff и многорукие бандиты

Поскольку рассматриваемая задача предполагает наличие большого количества классов в датасете, довольно естественно возникает ситуация, когда в тренировочном наборе данных не имеются представители всех классов датасета. Если при этом оптимизируется метрика общей точности модели на поступающих данных, то возникает проблема выбора - для нас выгоднее улучшать точность на уже известных классах за счет отбора образцов, близких к таковым в обучающей выборке (exploitation) или же попытаться найти в датасете семплы, которые повысят нашу предсказательную способность на малоизвестных или совсем неизвестных классах (exploration).
Данную задачу, а также в общем случае выбор оптимального алгоритма активного обучения на каждом шаге можно рассматривать как задачу о многоруких бандитах, предполагая, что награда, получаемая от каждого из алгоритмов не меняется полностью после каждого этапа семплирования.

3.5 Задача о многоруких бандитах

Нам дано множество неизвестных распределений X и N попыток обращения к ним, при обращении к какому-либо распределению мы получаем число x_i, называемое "наградой". Наша задача заключается в получении максимальной суммарной награду за N попыток, не имея данных о распределениях X.

3.5.1 Семплирование Томпсона

Одним из методов решения задачи многоруких бандитов является семпирование Томпсона.
Предположим, что у нас есть неизвестные распределения наград X, тогда для каждого распределения мы можем задать функцию, вычисляющую апостериорную вероятность награды каждого из распределений $P(\mu|x_{i1}...x_{ij})$
где μ - награда за обращение к распределению X_i, а $x_{i1}, ..., x_{ij}$ - награда за все предыдущие попытки обращения к данному распределению.
Тогда на i-м шаге мы обращаемся к каждому из апостериорных распределений и выбираем X_j, которому соответствует, которому соответствует распределение с наибольшим полученным числом. После чего, мы получаем награду из неизвестного распределения X_j и корректируем на ее основе апостериорную вероятность. Далее повторяем для i+1-го шага.

3.5.2 Валидация алгоритмов

Простой оценки метрик модели на одном или нескольких датасетах может быть недостаточно, поэтому возникает задача оценки достоверности полученных результатов. Доверительные интервалы метрик могут значительно упростить данную задачу, для их построения может использоваться хорошо зарекомендовавший себя в прикладных задачах метод Bootstrapping, позволяющий при помощи семплирования с заменой элементов изначальной выборки строить доверительные интервалы для практически любых статистик этой выборки [18].

3.6 Схожие работы: XMLC

Extreme Multi-label Classification - задача, когда необходимо каждому объекту присвоить несколько лейблов из крайне большого пространства классов. В работе [25] рассматривается следующий подход активного обучения для XMC: классы разделяются на seen и unseen, после чего строится матрица схожести классов между собой (для всех классов) и происходит разметка выборки семплов на предмет принадлежности к классу из множества seen. После этого на основе схожести известных классов с неизвестными отбирается n наиболее схожих классов, после чего элементы обучающей выборки размечаются на принадлежность к этим классам.
Данный подход также относится к активному обучению и рассматривает ситуацию наличия большого количества разных классов, однако его нельзя перенести на нашу задачу, поскольку она оперирует понятиями, специфичными именно для multi-label classification.

3.7 Выводы

В данном разделе бы проведен аналитический обзор предшествующей литературы по извлечению признаков из текстов, классическим подходам машинного обучения. При этом работ, рассматривающих ограничения, совпадающие с таковыми у экстремальной многоклассовой классификации текстов найдено не было.

4 Постановка задачи

4.1 Предобработка датасета

На вход программы подается частично размеченный текстовый датасет с большим количество различных классов (более 100). Предполагается, что все возможные классы известны. Далее происходит предобработка (препроцессинг) текста: разделение на токены и нормализация с последующим переводом предложений в векторные представления (эмбеддинги), косинусное расстояние между любыми двумя парами которых отображает их семантическую близость. [1]

4.2 Задача активного обучения

После успешного получения векторных представлений текстов, перед нами появляется задача наиболее эффективной доразметки исходного датасета.

4.2.1 Формальная постановка задачи

Задача активного обучения выглядит следующим образом:

$$X = \{x_1, x_2, ..., x_n\},\ x_i \in R^{dims}$$

$$Y = \{class_1, class_2, ..., class_m\}$$

X - множество всех признаков объектов датасета
Y - множество всех различных классов
x_i - векторизованный текст элемента датасета
n - количество элементов датасета

$dims$ - размерность входного эмбеддинга предложения
m - количество различных классов в датасете

$$fO : X \longrightarrow Y$$

fO - oracle function - функция, восстанавливающее истинное значение метки класса по его векторному представлению

$$T \subset X,\ P \subset X,\ V \subset X$$

Где T, V - train, validation - подмноженства X, для которых известны значения $fO(x_i),\ x_i \in T\ or\ x_i in V$
P - pool - подмножество X, для которого значения fO неизвестны.
Введем переменную $Budget$ - максимальное количество объектов, которые можно переместить из множества P в множество T.

$$Model : X \longrightarrow Y$$

$$Fit : (t_i, y_i) \longrightarrow Model,\ where\ y_i \in fO(t_i)\ and\ t_i \in T$$

$Model$ - модель машинного обучения, приближающая значение функции fO для векторного представления текста из датасета.
Fit - алгоритм, вычисляющий $Model$ на парах $(t_i, fO(t_i)),\ t_i \in T$

4.2.2 Практическая постановка

Будем предполагать, что ресурсы на разметку датасета ограничены. Полное множество возможных классов известно нам заранее. Необходимо добиться максимального прироста качества предсказаний модели за счет разметки (получения лейблов) не более чем X членов неразмеченной выборки, где X определяется нашим бюджетом на разметку.

4.3 Web интерфейс

Веб интерфейс должен упростить взаимодействие с разработанным кодом. Предполагается, что пользователь сможет взаимодействовать с кодом алгоритма при помощи графического интерфейса после незначительной конфигурации входных данных
Помимо элементов для взаимодействия необходимо реализовать визуализацию самого датасета, а также кандидатов на разметку.

5 Предлагаемый подход

5.1 Подготовка датасета

Для решения задачи токенизации применяется униграмная модель из библиотеки SentencePiece [7], для лемматизации слов применена модель WordNet Lemmatizer из библиотеки NLTK [8]. Векторные представления слов получаются при помощи усреднения векторов слов внутри каждого текста.
Также были попытки использовать векторые представления, создаваемые сверточными нейросетями, обучаемыми на задачу классификации, а также моделью

MPNet, так как они давали гораздо большее качество итоговой модели классификации, однако оба подхода не удалось использоваться в дальнейшем из-за слишком больших вычислительных затрат на предобработку, а также размеров получаемых представлений.

5.2 Методы отбора новых элементов для разметки

В основном для семплирования новых кондидатов применяются: least confidence, margin, entropy sampling [15]
Для отбора кандидатов на основе несогласия коммитета моделей[14] предполагалось использовать SVM, логистическую регрессию и KNN, однако SVM не подходит для обучения на больших размерах даннных (квадратичная зависимость времени обучения от размера обучающей выборки), поэтому пришлось отложить данный классический метод ранжирования из-за его плохой масштабируемости. Также для уменьшения количества выбросов среди кандидатов, была применена метрика информационной плотности области вокруг объекта, вычисляемая апроксимацией метрики на 100 ближайших соседях (так как вычислять метрику честно считая попарные расстояния всех объектов не представляется возможным на больших датасетах). Она совмещается вместе с уже существующими методами за счет добавления с определенным коэффициентом в итоговый рейтинг или подмешивания семплов выбранных по критерию близости к известным данным в наборы для разметки. Предполагается, что эти действия будут помогать более оптимально решать проблему exploration-exploitation tradeoff.
Однако предложенный подход скорее всего не будет оптимальным - он не предполагает, что в каких-то случаях учет определенных алгоритмов семплирования (например, учет предсказаний модели при недостаточном количестве семплов в обучающей выборке) могут оказывать негативное влияние на предсказание ценности объектов для модели. Получается, что мы хотим на каждом этапе динамически определять лучший алгоритм семплирования.
Данную задачу можно свести к задаче многоруких бандитов, предполагая, что одни и те же методы могу оказываться оптимальными на протяжении множества итераций подряд. Изучение нескольких статей по данной тематике [33, 34] позволило выбрать семлирование Томпсона как наиболее многообещающий алгоритм решения данной задачи, а в качестве функции награды было выбрана

$$rew_i = \begin{cases} \sqrt[3]{(truePos_i - truePos_{i-1})^2} & \text{if } truePos_i - truePos_{i-1} \geq 0 \\ truePos_i - truePos_{i-1} & \text{otherwise} \end{cases}$$

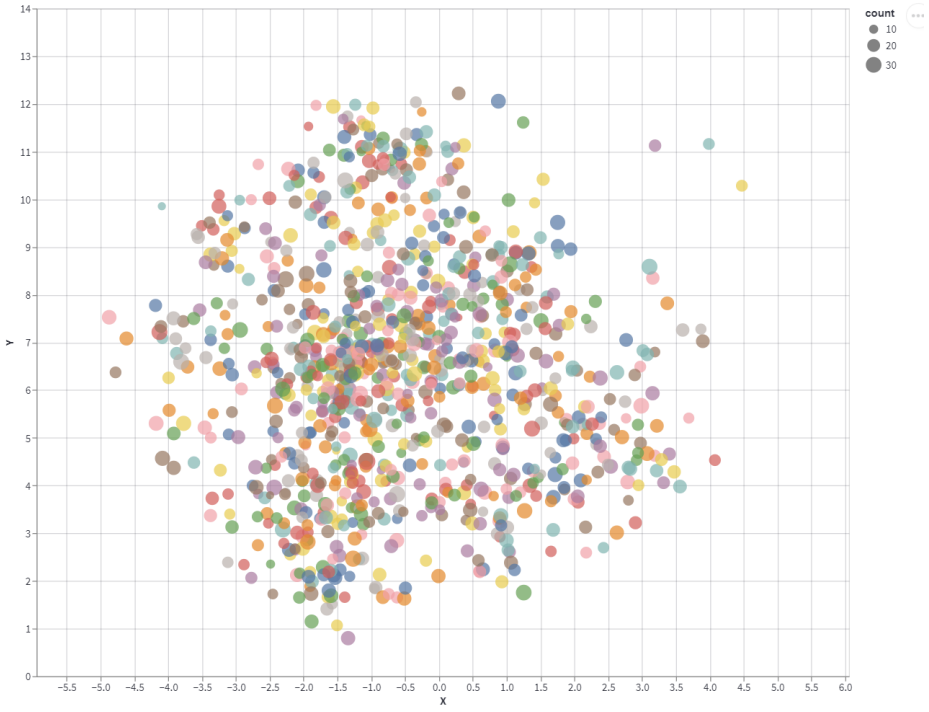
$truePos$ - количестве верно предсказанных классов объектов валидационной выборки. Такое уравнение подобрано на основе эмперически более быстрого определения семплированием лучшего бандита.

5.3 Web интерфейс

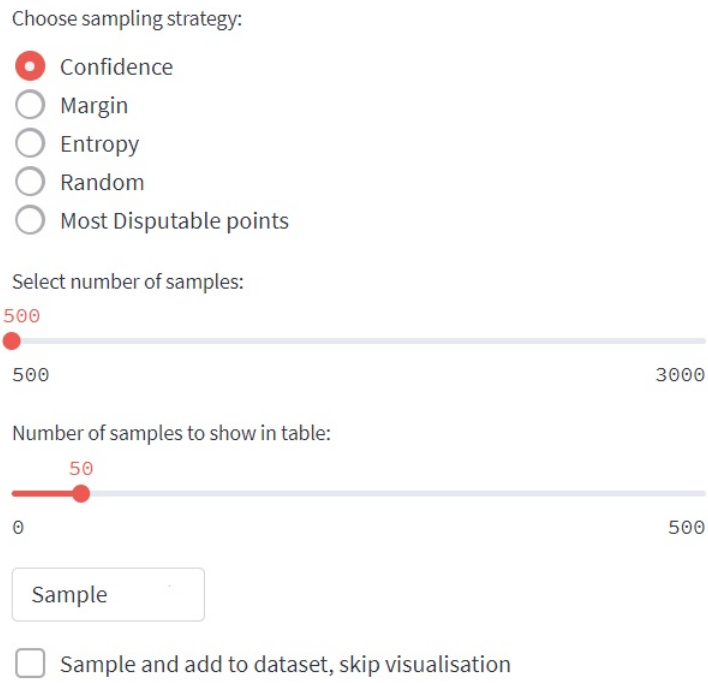
Веб приложение предполагает интерактивную демонтстрацию работы подхода. Оно выполнено с использованием библиотеки Streamlit [26]. В то время как визуализация самого датасета производится при помощи проекции на плоскость датасета алгоритмом UMAP [23] с последующей группировкой по классу усреднением координат.

5.4 Контроль качества экспериментов

Чтобы получать доверительные интервалы для метрик датасет семплируется с заменой в полном размере 1000 раз, затем для всех новых выборок считается целевая метрика и берется ее 0.05 и 0.95 перцентили, соответствующие 95 уровню доверия.



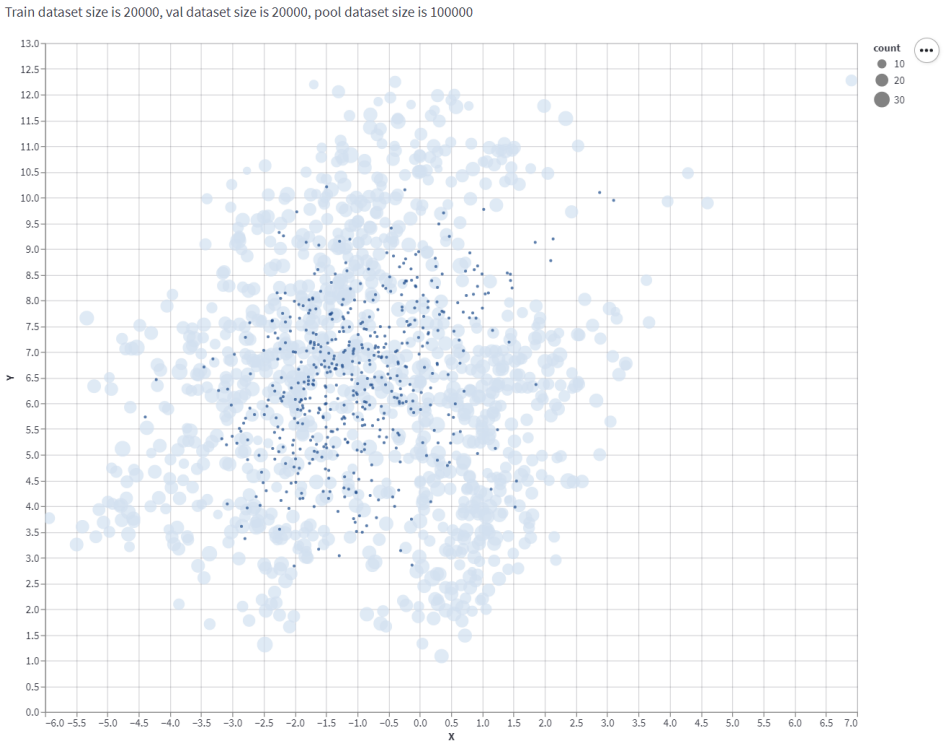
Пример отображения части датасета на плоскость



Текущий функционал семлирования, доступный пользователю

	id	title	selftext
945059	4zwvj7	No offence, but I think you're telling porkies...	Actually no I don't, but I do wonder about the honesty of some of the tales in here sometimes. I'm reading post after post of success stories with jumping. Even when people are thinking it doesn't work they post afterwards, all overjoyed that it works. Yet here I am. I only tried to change something small, ie to get back together with my partner... by small I mean not outside the realm of possibility, as we've split a few times before and got back together. I did the glasses method and it's been two weeks. <lb><lb>I would love to be sitting here typing about how thrilled I am that it's worked, but no, nothing. Not even a hint of it. <lb><lb>Maybe I did something wrong with the experiment. Maybe I should try again. Maybe I'm the worst person in the world and he'll never even speak to me again. It's not even that important in the grand scheme of things, but I really want this experiment to work in general, yet I can't even get past the tester run.<lb><lb>Is there anyone else here who can't get this to work?
30992	59lrc2	[771] Super Man	So I have been back on this sub and doing some critiques. I felt it was only fair to show some of my [very rough] writing so I wasn't sitting in an ivory tower.<lb><lb>This is the intro to an idea I've been playing with very recently about a very delusional teenager who ends up thinking he's essentially a hero that's called to save this girl he's into from a gang. I love messing with unreliable and otherwise distasteful narrators, and the tone of this piece is so casual compared to the other stuff I'm working on at the moment that it's a nice break.<lb><lb>I think it may just be boring shit, but I'm not sure. Let me know!<lb><lb>EDIT/Disclaimer: If you care much about strict grammar when it comes to sentence structure, this will not be your jam.<lb><lb>https://docs.google.com/document/d/1-gCkJf2sj1sCNIngx3s7RdL9CfpIsgym0AqmK3dZ0VY/edit?usp=sharing
			I've found many major MSP's using ConnectWise, which leads me to believe it's a solid product. If it can work for them, then it should be good for me to start with, right?<lb><lb>This is where it scares me... I go to the CW website and watch the video. But I have no idea what all the products are. Manage. Automate. Sell.

Превью семлированных элементов датасета



Визуализация семлированных точек датасета, нанесенных на центры классов основного датасета.

6 Эксперименты

6.1 Исходные данные

В качестве датасета был выбрал датасет The reddit self-post classification task [4]. Он содержит 1013 классов, по 1000 примеров каждый. При этом данные в нем зашумлены и не являются идеальными научными текстами (как например в DBPedia [5]). Также каждый элемент датасета представлен в среднем текстом размера 200 слов, что является разумным с вычислительной точки зрения в сравнении с датасетом Amazon Product Browse Node Classification Data [6], в котором

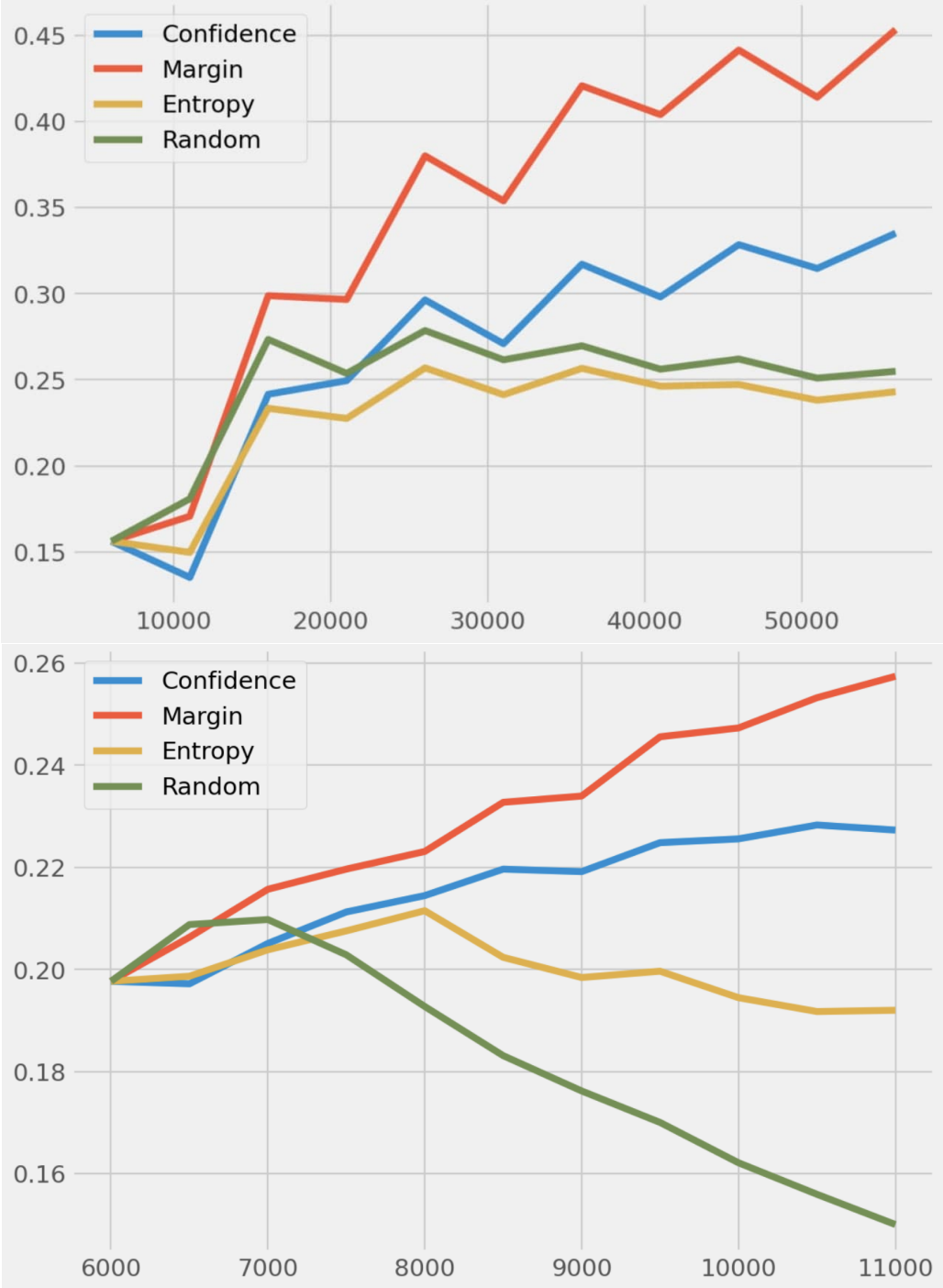
каждый образец привязан к большому количеству текстовой информации, в разы превышающую объем таковой в selfpost.

6.2 Промежуточные результаты

На данным момент тестирование было проведено на выбранном датасете [4] с использованием модели логистической регрессии. Семплирование проводилось при помощи методов least confidence, margin и entropy на вероятностях, предсказанных моделью.
Также была попытка использовать для разметки нейросетевые подходы, однако в силу вычислительной сложности и размера датасета от идеи пришлось отказаться в пользу более простых моделей.

6.2.1 Простые методы семлирования

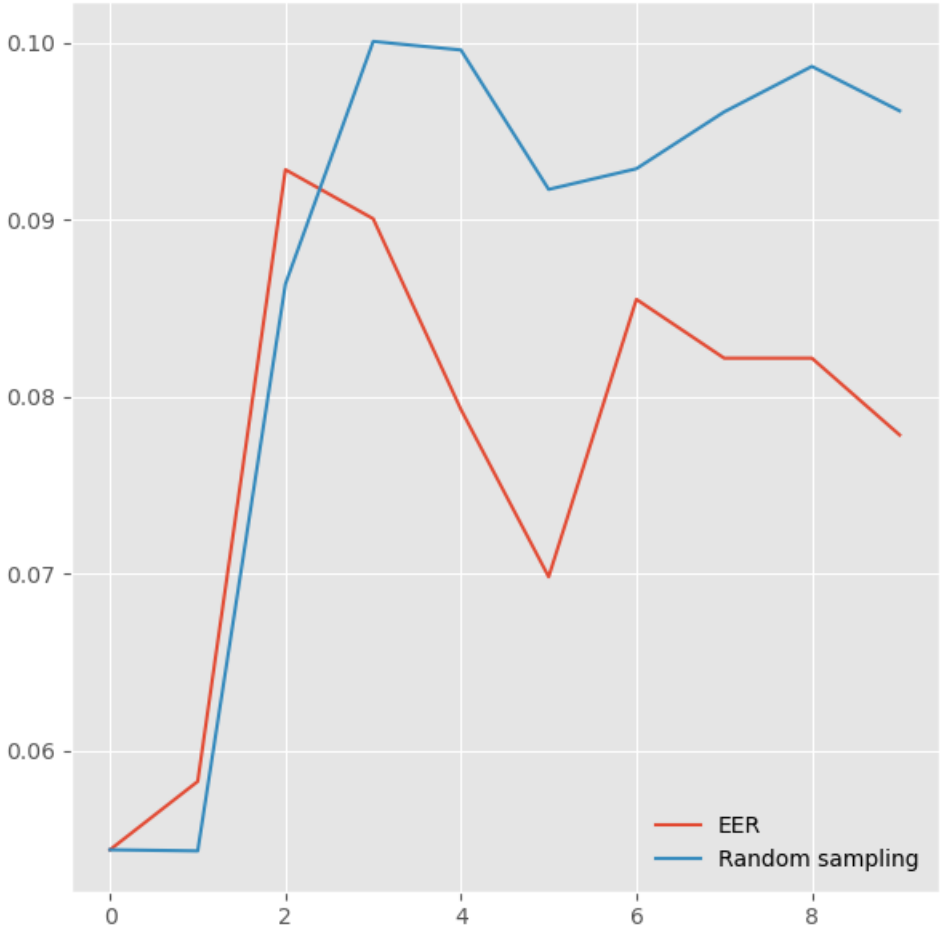
Семплирование методом margin превзошло все остальные виды семплирования (+30 процентов accuracy против прироста в 14% и 9% для least confidence и entropy соответственно). Малый прирост точности для семплирования по энтропии может быть связан с недостаточной для данного метода предсказательной способностью модели логистической регрессии.



Зависимость роста accuracy от размера обучающей выборки с разными алгоритмами отбора объектов с соотношением выборок $train : validation : pool$ 60000 : 40000 : 900000 и 6000 : 4000 : 990000

6.2.2 Expected Error Reduction

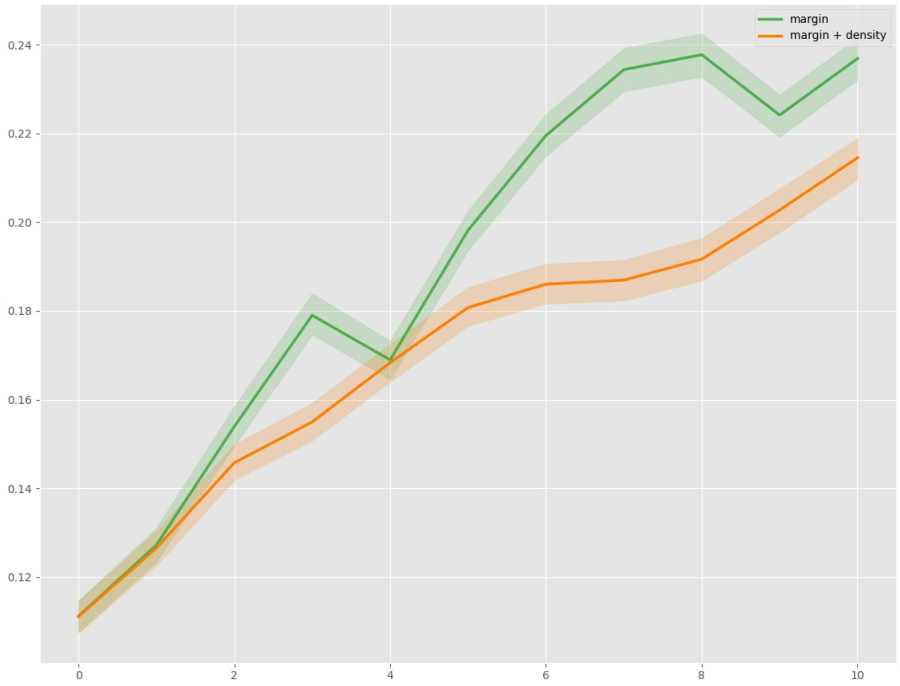
Также был протестирован метод отбора кандидатов на разметку, оценивающий ожидаемое уменьшение ошибки модели. Метод требует крайне высоких вычислительных затрат, поэтому изначально тестировался на небольших датасетах с малым количеством классов. Были использованы 3 датасета: banknote authentication Data Set [29], Wine Quality Data Set [30], Abalone Data Set [31]. В результате было принято решение не использовать метод, так как он не приносит заявленного в статьях-источниках качества и вычислительная сложность слишком высока (приходится переобучать модель для каждого отбираемого объекта $|distinct(y)|$ раз)



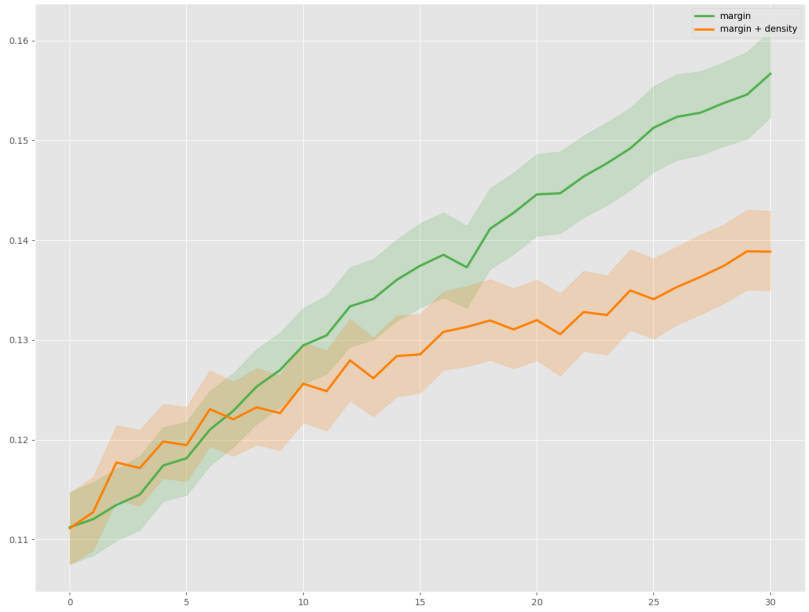
Зависимость точности модели классификации от количества добавленных объектов, выбранных двумя разными методами.

6.2.3 Плотность информации

Следующим тестировался подход добавления к коэффициенту ранжированию семплов множетеля, учитывающего плотность объектов в его окресности, которая приближалась средним расстоянием до 100 ближайших соседей точки.



Зависимость точности модели классификации от номера итерации семплирования. Начальный размер train - 1000 объектов, на каждой итерации добавлялось по 1000 объектов

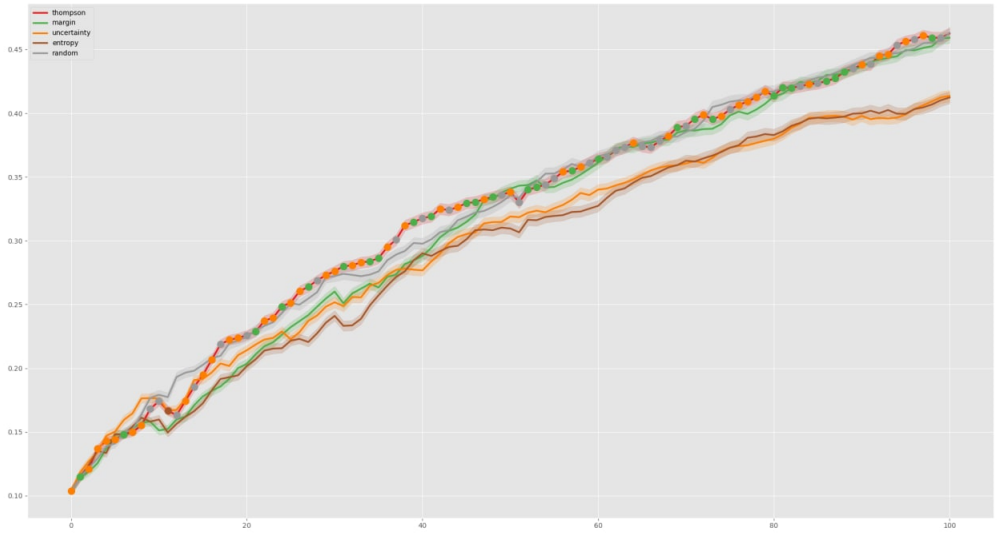


Зависимость точности модели классификации от номера итерации семплирования. Начальный размер train - 1000 объектов, на каждой итерации добавлялось по 50 объектов

Получается, что добавление учета информационной плотности в семплирование на данном датасете значительно ухудшает эффективность отбора объектов для разметки, а не улучшает ее благодаря борьбе с выбросами, как это ожидалось. Возможно это связано с высокой средней информативностью всех объектов или наличия классов с достаточно разреженными представителями.

6.3 Дальнейшие исследования

Также предпринимались попытки сделать алгоритм жадного онлайн подбора метода фильтрации кандидатов. Он имеет следующий вид:
4 возможных алгоритма семплирования - margin sampling в паре с information density, entropy sampling в паре с information density, random sampling в паре с information density и query by committee disagreement в паре с information density. Выбор между ними осуществляется при помощи семплирования Томпсона.
Для оценки достоверности результатов для метрик посчитаны доверительные интервалы при помощи методики bootstraping. Используется 1/10 часть выбранного изначально датасета reddit self post.



Зависимость точности модели классификации от итерации семлпирования - на первой итерации в train было 1000 объектов и далее добавлялось по 10 объектов каждую итерацию. Точки на красном графике соответствуют выбронному алгоритмом методу (см. легенду)

Как видно из графика, хоть в выбираемых семплированием Томпсона алгоритмах просматриваются определенные закономерности и постоянство, оно не смогло превзойти по качество бейзлайновые методы (эта проблема станет объектом будущих исследований автора).

6.4 Итоги тестирования методов

Подход	Результат
Uncertainty sampling	Подход показывает заметно более быстрый прирост качества модели в сравнении с случайным отбором кандидатов
Information density	Учет метрики плотности ухудшает качества модели на 30 процентов
EER	Попытки оценивать ожидаемое уменьшение ошибки не принесли результата даже на модельных датасетах.
Динамической подбор алгоритма семплирования	Алгоритм не дал улучшения качества модели, однако он показал, что способен выучиться опредленные закономерности и, возможно, он мог бы показать положительные результаты при определенной доработке.

7 Заключение

Автор провел обзор литературы из области активного обучения, которая дает общее представление о задаче, путях ее решения и проблемах, с которыми приходится сталкиваться при ее выполнении.
В явном виде рассмотрение задачи активного обучения для задачи экстремальной классификации коротких текстов не встречается в научной литературе. Однако существует множество подходов к задаче активного обучения в целом, которые были рассмотрены и протестированы на применимость к данной задаче подходы.
Задача исследования заключалась в анализе и тестировании методов активного

обучения на задаче экстремальной многоклассовой классификации и выявлении наиболлее эффективных подхов к решению задачи разметки датасета.
Как видно из проведенного анализа алгоритмы активного обучения не всегда оказываются применимыми к экстремальной классификации, а также иногда требуют значительных вычислительных оптимизаций (например, алгоритм вычисления информационной плотности)
На данный момент наиболее эффективными показали себя классические методы семплирования на основе неуверенности модели, в то время как другие методы оказались неэффективными, либо нуждались в доработке.
Примером метода нуждающегося в доработке является жадного онлайн подбора лучшего алгоритма семлирования, основанный на задаче о многоруких бандитах. Данный подход в текущей своей реализации не дал улучшения качества относительно применяемых по-одиночке подходов, однако показал способность неслучайно выбирать наиболее ожидаемо выгодный метод семплирования.

Список литературы

[1] T. Mikolov, I. Sutskever, K. Chen: Distributed Representations of Words and Phrases and their Compositionality // [Электронный ресурс] arXiv.org. Режим доступа: https://arxiv.org/abs/1310.4546 свободный. (дата обращения: 01.02.2022)

[2] P. Bojanowski, E. Grave, A. Joulin, T. Mikolov Enriching Word Vectors with Subword Information // [Электронный ресурс] arXiv.org. Режим доступа: https://arxiv.org/abs/1607.04606 свободный. (дата обращения: 08.02.2022)

[3] J. Pennington, R. Socher, C. D. Manning GloVe: Global Vectors for Word Representation // [Электронный ресурс] Режим доступа: https://nlp.stanford.edu/pubs/glove.pdf свободный. (дата обращения 08.02.2022)

[4] M. S. Jones The reddit self-post classification task // [Электронный ресурс] Kaggle.com. Режим доступа: https://www.kaggle.com/mswarbrickjones/reddit-selfposts/?select=subreddit_info.csv свободный. (дата обращения: 11.2021)

[5] DBPedia Classes
DBpedia https://www.dbpedia.org/ // [Электронный ресурс] Kaggle.com. Режим доступа: https://www.kaggle.com/danofer/dbpedia-classes свободный. (дата обращения: 11.2021)

[6] S. Surana Amazon Product Browse Node Classification Data // [Электронный ресурс] Kaggle.com. Режим доступа: https://www.kaggle.com/subhamjain/amazon-product-browse-node-classification-data свободный. (дата обращения: 11.2021)

[7] SentencePiece / Google Open Source // [Электронный ресурс] Github.com. Режим доступа: https://github.com/google/sentencepiece свободный. (дата обращения 02.02.2022)

[8] Natural Language Toolkit // [Электронный ресурс] https://www.nltk.org/. Режим доступа: свободный. (дата обращения 02.02.2022)

[9] word2vec / Google research // [Электронный ресурс] Google Code Archive code.google.com. Режим доступа: https://code.google.com/archive/p/word2vec/ свободный. (дата обращения: 11.2021)

[10] FastText Word vectors for 157 languages / Facebook Open Source // [Электронный ресурс] Fasttext.cc. Режим доступа: https://fasttext.cc/docs/en/crawl-vectors.html свободный. (дата обращения: 11.2021)

[11] O. Poltavets GloVe Reddit Comments // [Электронный ресурс] Kaggle.com. Режим доступа: https://www.kaggle.com/leighplt/glove-reddit-comments свободный. (дата обращения 11.2021)

[12] Support Vector Machine Sklearn documentation // [Электронный ресурс] Sklearn documentation scikit-learn.org. Режим доступа: https://scikit-learn.org/stable/modules/svm.html свободный. (дата обращения 02.02.2022)

[13] Pytorch python framework // [Электронный ресурс] PyTorch.org. Режим доступа: свободный. (дата обращения 02.02.2022)

[14] H.S. Seung, M. Opper, and H. Sompolinsky: Query by committee // In Proceedings of the ACM Workshop on Computational Learning Theory, pages 287–294 – 1992.

[15] D. Lewis and W. Gale. A sequential algorithm for training text classifiers // In Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval, pages 3–12 - 1994.

[16] B. Settles. Active Learning Survey // volume 6 of Synthesis Lectures on Artificial Intelligence and Machine Learning.Morgan & Claypool - 2012

[17] T. Kudo Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates // [Электронный ресурс] arXiv.org. Режим доступа: https://arxiv.org/abs/1804.10959 свободный. (дата обращения: 06.02.2022)

[18] Bootstrapping (statistics) // [Электронный ресурс] Wikipedia - The Free Encyclopedia. Режим доступа: https://en.wikipedia.org/wiki/Bootstrapping_(statistics) свободный. (дата обращения 07.02.2021)

[19] C.E. Shannon: A mathematical theory of communication // Bell System Technical Journal, 27:379–423,623–656 – 1948

[20] H.S. Seung, M. Opper, and H. Sompolinsky. Query by committee. In Proceedings of the ACM Workshop on Computational Learning Theory, pages 287–294 - 1992.

[21] Z. Xu, R. Akella, Y. Zhang Incorporating Diversity and Density in ActiveLearning for Relevance Feedback, / University of California, Santa Cruz, CA, USA, 95064 // [Электронный ресурс] Режим доступа: https://users.soe.ucsc.edu/~yiz/papers/c8-ECIR07.pdf свободный (дата обращения 07.02.2022)

[22] H. T. Nguyen, A. Smeulders Active Learning Using Pre-clustering / Intelligent Sensory Information Systems, University of Amsterdam, Faculty of Science, Kruislaan 403, NL-1098 SJ, Am-sterdam, The Netherlands // [Электронный ресурс] Режим доступа: https://icml.cc/Conferences/2004/proceedings/papers/94.pdf свободный. (дата обращения 07.02.2022)

[23] L. McInnes, J. Healy, J. Melville UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction // [Электронный ресурс] arXiv.org. Режим доступа: https://arxiv.org/pdf/1802.03426 свободный. (дата обращения: 07.02.2022)

[24] A. Coenen, A. Pearce Understanding UMAP // [Электронный ресурс] Режим доступа: https://pair-code.github.io/understanding-umap/ свободный (дата обращения 07.02.2022)

[25] Sihong Xie, Philip S. Yu, Active zero-shot learning: a novel approach to extreme multi-labeled classification // [Электронный ресурс] Режим доступа: https://link.springer.com/content/pdf/10.1007/s41060-017-0042-5.pdf свободный. (дата обращения 08.02.2022)

[26] Streamlit python library // [Электронный ресурс] Режим доступа: https://streamlit.io/ свободный. (дата обращения 08.02.2022)

[27] A. Vaswani, N. Shazeer, N. Parmar and others Attention Is All You Need // [Электронный ресурс] Режим доступа: https://arxiv.org/abs/1706.03762 свободный. (дата обращения 15.02.2022)

[28] N. Roy, A. McCallum Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction // [Электронный ресурс] Режим доступа: https://groups.csail.mit.edu/rrg/papers/icml01.pdf свободный. (дата обращения 15.04.2022)

[29] banknote authentication Data Set // [Электронный ресурс] Режим доступа: https://archive.ics.uci.edu/ml/datasets/banknote+authentication свободный. (дата обращения 15.04.2022)

[30] Wine Quality Data Set // [Электронный ресурс] Режим доступа: https://archive.ics.uci.edu/ml/datasets/wine+quality свободный. (дата обращения 15.04.2022)

[31] Abalone Data Set // [Электронный ресурс] Режим доступа: https://archive.ics.uci.edu/ml/datasets/abalone свободный. (дата обращения 15.04.2022)

[32] K. Song, X. Tan, T. Qin MPNet: Masked and Permuted Pre-training for Language Understanding // Электронный ресурс Режим доступа: https://www.microsoft.com/en-us/research/publication/mpnet-masked-and-permuted-pre-training-for-language-understanding/ свободный. (дата обращения 24.04.2022)

[33] Steve Roberts Bandit Algorithms Multi-Armed Bandits: Part 3 // [Электронный ресурс] Режим доступа: https://towardsdatascience.com/bandit-algorithms-34fd7890cb18 свободный. (дата обращения 27.04.2022)

[34] Steve Roberts Thompson Sampling Multi-Armed Bandits: Part 5 // [Электронный ресурс] Режим доступа: https://towardsdatascience.com/thompson-sampling-fc28817eacb8 свободный. (дата обращения 27.04.2022)