Московский Физико-Технический Институт (Государственный Университет)

Факультет Управления и Прикладной Математики

Кафедра Интеллектуальные Системы

ДИПЛОМНАЯ РАБОТА МАГИСТРА

«Классификация эмоциональной окраски сообщений в социальных сетях»

Выполнил:

студент 6 курса 774 группы

Савинов Николай Анатольевич

Научный руководитель:

д.ф.-м.н., профессор

Воронцов Константин Вячеславович

Заведующий кафедрой

Интеллектуальные Системы,

член-корреспондент РАН _____ К. В. Рудаков

Москва, 2013

Содержание

Аннотация

В крупных компаниях часто возникает задача анализа реакции пользователей на продукты этих компаний. Одним из источников данных для этой задачи может выступать Твиттер. В данной работе производится исследование и экспериментальное сравнение возможных подходов к анализу тональности сообщений про компанию в Твиттере. Полученные результаты показывают важность использования синтаксических признаков и мета-информации из Твиттера в дополнение к стандартным униграммным признакам. Из результатов также следует, что возникающая задача классификации должна решаться методами, устойчивыми к несбалансированности классов в обучающей выборке и наличию полудублей — почти совпадающих сообщений. Согласно экспериментам, лучшим алгоритмом из исследованных является логистическая регрессия, использующая синтаксические признаки, мета-информацию из Твиттера и методы работы с несбалансированными выборками.

1 Введение

Классификация эмоциональной окраски сообщений пользователей относительно продуктов компании важна для определения достоинств и недостатков этих продуктов. Такая классификация может обеспечить обратную связь с пользователями и позволяет спланировать именно те улучшения продуктов, которые действительно необходимы.

Задача классификации эмоциональной окраски сообщений в социальной сети относительно некоторого компании формулируется следующим образом. Дано множество всех сообщений Твиттера $T = \{m_i\}_{i=1}^{N}$. Среди этого множества требуется выделить подмножество сообщений $C = \{m_j\}_{j=1}^{K} \subseteq T$, относящихся к компании и ее продуктам. Также требуется подмножество C классифицировать на три группы сообщений: положительные (метка 1), отрицательные (метка -1) и нейтральные (метка 0). Таким образом, множеством меток классификации является $Y = \{1, -1, 0\}$. Правильная классификация определяется человеком-асессором, который присваивает сообщению m_j метку $y_j \in Y$ на основе выданной ему инструкции и личного восприятия сообщения. Качество выделения класса C не рассматривается в данной работе, способ выделения фиксируется некоторым разумным образом. Метрики качества классификации множества C на три класса описываются в разделе 1.4.

Классический метод решения поставленной задачи, описанный в [1], предполагает наличие двух этапов. На первом этапе с помощью методов информационного поиска собираются сообщения из Твиттера, потенциально имеющие отношение к компании или ее сервисам. После этого наступает второй этап, во время которого производится анализ тональности сообщений и выделяются те из них, которые несут положительную или отрицательную окраску относительно компании.

Возможным решением первой части задачи является использование поисков по социальным медиа. Для русскоязычных пользователей главным поставщиком такой услуги является сервис “Поиск по Блогам” [2] компании Яндекс. Таким образом, задача сводится к формулировке запроса, охватывающего отзывы как про саму компанию, так и про ее сервисы. Важной особенностью является наличие слэнговых синонимов, употребляемых пользователями по отношению к этим сервисам. Таким образом, от аналитика, составляющего подобный запрос, требуется знание языка

пользователей Твиттера. В данной работе поисковый запрос фиксируется и дальнейшее исследование его оптимальности производиться не будет.

Вторая задача имеет несколько распространенных решений. Подробный обзор возможных подходов дан в [3]. Следует отметить, что в практически работающих приложениях чаще применяются системы правил [4], в то время как в научной среде большую популярность имеют методы машинного обучения, среди которых ключевые места по мнению исследователя [5] занимают линейные и обобщенно-линейные методы классификации SVM и логистическая регрессия.

Недавния публикация [6] показывает эффективность двухстадийного иерархического подхода к решению задачи классификации тональности сообщений в Твиттере. Подход заключается в отделении на первой стадии тональных сообщений от нейтральных, а на второй — в разделении тональных на положительные и отрицательные. Именно эта идея построения иерархии из двух классификаторов исследуется в данной работе.

Следует отметить, что попытка применения указанного алгоритма исследователями [7] к реальным данным показывает, что F1-мера по классу тональных сообщений на первом этапе оказывается очень низкой. Авторы этой работы указывают на смещенность реального потока данных в сторону нейтральных сообщений. По-видимому, именно это смещение не позволяет достичь высоких показателей F1-меры по тональным сообщениям простыми линейными методами. Обзор [3] также подтверждает сложность отделения нейтральных сообщений от тональных, подчеркивая при этом относительную простоту второго этапа классификации. Поскольку именно тональные сообщения представляют практическую ценность, данная дипломная работа фокусируется на исследовании методов повышения точности и полноты по этому классу сообщений. Также приводятся результаты по качеству для второй стадии классификации, которая разделяет тональные сообщения на положительные и отрицательные.

В качестве эталона для сравнения в данной работе применяется метод логистической регрессии по признакам на внутрисловных побуквенных 4-граммах. Проведенное исследование включает сбор данных для машинного обучения, воспроизведение эталона и последовательное улучшение показателей точности и полноты за счет

использования различных методов машинного обучения и обработки естественных языков. По итогам экспериментов разработан метод эвристического формирования обучающей выборки на основе распределения Гиббса, устойчивый к наличию в выборке дубликатов, и метод порождения признаков классификации для алгоритма логистической регрессии, учитывающий синтаксический разбор (который представляет собой направленный граф синтаксических связей между словами в предложении), выделение отрицаний, метки тональности отдельных слов, части речи, эмотиконы (пиктограммы, выражающие эмоцию), а также мета-информацию из Твиттера. Кроме того, в работе производится сравнение методов решения проблемы несбалансированности выборки и приводятся графики зависимости качества от параметров алгоритма.

Разработанные методы были протестированы на реальных данных по тональности относительно компании Яндекс и показали свою эффективность по сравнению с эталонным методом.

1.1 Получение сообщений из Твиттера

Первоначальной задачей является формулировка запроса к сервису “Поиск по Блогам”, который учтет не только вхождения названия компании, но и покроет все ее возможные продукты. Дополнительную сложность добавляет тот факт, что некоторые названия могут состоять из нескольких частей, каждая из которых может употребляться в другом значении (не относящемся к компании или продукту). Например, для Яндекса таким сложным примером будет “я бар”, под которым пользователи обычно имеют в виду приложение Яндекс.Бар. Решить проблему помогает использование возможностей языка запросов информационного поиска, который позволяет ограничивать расположение ключевых слов в поисковой выдаче.

Более подробно, язык запросов [8] позволяет учесть следующую информацию, важную для рассматриваемой задачи:

1. Диапазон расстояний между словами.
2. Ограничение вхождений одним предложением.
3. Обязательное вхождение некоторого слова (без этого оператора современные поисковые системы ставят неявное “ИЛИ” между словами из запроса, [9]).
4. Учет точной формы слова (по умолчанию может применяться стемминг и лемматизация).

Как уже было упомянуто ранее, еще одну проблему составляют пользовательские синонимы официальным названиям компании и продукта. Из одних только синонимов компании Яндекс было собрано 9 слов.

1.2 Сбор данных для машинного обучения

Ключевой проблемой эффективного использования машинного обучения является сбор достаточного количества данных с метками классов. При этом желательным требованием является низкая стоимость разметки в расчете на одно сообщение. Для рассматриваемого иерархического подхода сложность получения такой разметки существенно различается в зависимости от подзадачи. Далее излагается сначала простая подзадача (отделение положительных сообщений от отрицательных), затем сложная (отделение тональных сообщений от нейтральных).

1.2.1 Подзадача положительный/отрицательный

В Internet есть несколько источников информации, которую можно преобразовать в размеченную выборку. К ним относятся:

1. http://films.imhonet.ru/
2. http://market.yandex.ru/
3. http://www.kinopoisk.ru/
4. http://amazon.com/
5. http://www.rottentomatoes.com/

Первые три источника относятся к русскому языку (и в первую очередь представляют практический интерес), остальные приведены для полноты картины. Все эти источники предоставляют отзывы пользователей с оценками. После проведения отсечения оценки по порогу появляются два класса (положительные — выше порога, и отрицательные — ниже). Достаточно простого поискового робота для скачивания необходимого количества отзывов.

При таком подходе классификатор, обученный на одних данных, будет в реальных условиях работать с другим потоком по другой тематике. В литературе такая постановка задачи называется transfer learning и domain adaptation. Эксперименты [10] показывают, что у задачи переноса обучения есть эффективное решение. Это решение заключается в построении признаков классификации, которые являются

функциями над исходными униграммами и учитывают совместную встречаемость слов.

Более того, для некоторых вариантов переноса можно использовать “наивную” схему, взятую авторами статьи в качестве эталона (для обучение классификатора по униграммам используется одно множество, для тестирования применяется другое; признаками классификации являются униграммы из пересечения). К примеру, таким свойство обладает перенос с темы “Кухонные электроприборы” (Kitchen appliances) на тему “Электроника” (Electronics) для данных с интернет-магазина Amazon. Отсюда можно сделать неформальное заключение, что если лексика двух областей схожа, перенос классификатора тональности можно осуществлять тривиально. В дальнейшем данные для обучения выбираются максимально близкими к целевой области, вопрос оптимальности “наивного” переноса не исследуется (ввиду наличия хороших по качеству результатов).

Поскольку конечной целью построения алгоритма в данном исследовании является классификация отзывов про компанию Яндекс, было принято решение использовать данные с Яндекс.Маркета. Мотивацией к использованию этих данных является сходство лексики по IT-тематике. Данные состоят из отзывов пользователей с проставленной оценкой от 1 до 5 и описанием достоинств и недостатков. Они были получены следующим образом: с Яндекс.Маркета были собраны примерно 600000 отзывов про электронику и бытовые приборы, порог отсечения был взят равным оценке 3, все меньше считалось негативными отзывами, все больше — позитивными, примеры с оценкой 3 выбрасывались из данных. Далее в качестве положительных примеров были взяты достоинства из положительных отзывов и недостатки из отрицательных.

1.2.2 Подзадача нейтральный/тональный

На данный момент неизвестны способы сбора дешевых данных для этой подзадачи. Исследователи [11] использовали идею представления отрывков сюжетов фильмов с сайта rottentomatoes в качестве примеров нейтральных сообщений, при этом в качестве тональных сообщений использовались отзывы пользователей. Следует отметить потенциальную опасность такого подхода: классификатор может научиться отличать источники данных, а не классы. Интуитивно, данные по всем классам

должны быть из одного источника. Достичь этого можно только с помощью ручной разметки. Было размечено три выборки:

1. Размером 1000 четырьмя асессорами (с полным перекрытием: каждое сообщение размечалось всеми асессорами),

2. Размером 6000 двумя асессорами (с небольшим перекрытием: подмножества размечаемых асессорами сообщений пересекались по небольшой доле сообщений),

3. Размером 26000 одним экспертом.

Далее первая выборка будем обозначаться tweets1000, вторая — tweets6000,третья — tweets26000. Следует отметить, что свойства выборок несколько различаются, поскольку для первых двух выборок применялась фильтрация по рейтингу автора сообщения (отсекались сообщения авторов с низким рейтингом).

Полудубли. Для всего потока сообщений была произведена фильтрация полудублей. Фильтрация состояла из нескольких этапов:

1. Твит разделялся на части, каждая из которых принадлежит одному автору (некоторые твиты представляют собой цепочку ретвитов и ответов на сообщения других пользователей, эта цепочка растет справа налево).

2. Из твита удалялись html-ссылки, хэштеги и html-разметка.

3. Производилась нормализация пробельных символов (после нормализации слова разделяются только одиночными проблелами).

4. Объединялись тексты полученных частей и оставлялись только уникальные.

По итогам такой фильтрации доля полудублей (сообщений, которые можно выбросить из-за наличия похожих) составляет 42%. Как будет показано далее, такая значительная доля полудублей затрудняет использование некоторых методов машинного обучения.

Согласованность асессоров. Для выборки tweets1000 была оценена согласованность асессоров. Асессорам предлагалось разметить твиты на три класса: положительные, отрицательные и нейтральные. Будем называть согласованность полной, если все асессоры одинаково оценили тональность записи. Частичной согласованностью считается ситуация, в которой большинством голосов можно установить тональность, но полной согласованности нет. Остальные случая назовем отсутствием согласованности. По итогам проведенного исследования, доля случаев с полной согласованностью составляет 30%, с частичной — 55%, с отсутствием согласованности — 15%. Таким образом, задача является сложной даже для человека, и ожидать высоких результатов от применения машинного обучения не следует.

1.3 Машинное обучение в классификации текстов и задаче анализа тональности

Стандартным подходом к задаче классификации текстов является использование линейных методов SVM и логистической регрессии на униграммных признаках. Их популярность можно объяснить особенностями задачи:

- Для классификации текстов часто используются выборки, содержащие миллионы объектов и признаков. В частности, второй этап классификации в предложенной схеме анализа тональности использует такие массивные выборки. Поэтому скорость и масштабируемость обучения алгоритма являются важными факторами в данной задаче. При этом для обучения линейных методов разработаны эффективные процедуры, позволяющие работать с большими выборками. Одним из примеров является пакет машинного обучения liblinear, который показывает значительный выигрыш во времени работы по сравнению с некоторыми другими менее специализированными пакетами согласно [12]. Другим известным примером является пакет Vowpal Wabbit [13], предоставляющий возможности распараллеливания обучения на вычислительном кластере с помощью технологии Hadoop AllReduce [14]. Вышеуказанные примеры подчеркивают предпочтительность использования линейных методов с точки зрения скорости обучения.

- Признаковое описание текста обычно представляет собой сильно разреженный вектор высокой размерности $D \sim 10^6$, но перемножение с вектором весов выполняется за $O(N)$, где N — число ненулевых элементов признакового описания (для Твиттера $N \sim 10^2$). Это быстрая операция по сравнению, например, с принятием решения на основе решающих деревьев в методе градиентного бустинга, если эти деревья используют все признаки. Таким образом, на этапе классификации линейные методы также более предпочтительны.

Опишем простой и эффективный алгоритм классификации текстов, который взят в данной работе в качестве эталона для сравнения. К каждом этапу приведены обоснования его использования.

1. Текст сообщения разбивается на слова, которые приводятся к нижнем регистру. Это стандартная операция в информационном поиске. Из неформальных соображений понятно, что она несколько снижает точность в данной задаче (иногда тональность выражается всеми заглавными буквами), но при этом значительно повышает полноту (пользователи социальных сетей не заботятся о корректном использовании заглавных букв, поэтому статистики по каждой форме оказывается меньше).

2. Выбрасывается пунктуация. Для короткого сообщения в Твиттере (длина ограничена 140 символами) это ничего не испортит, поскольку одно сообщение чаще всего несет одну мысль, и деление на предложения неважно. При этом следует отметить, что для длинных текстов (посты в блогах, например) это предположение не выполняется.

3. Выбрасываются цифры. Статистики для них слишком мало, чтобы использовать их в качестве признаков.

4. Из полученных слов строятся внутрисловные 4-граммы (захватывающие пробел до и после слова). Это стандартное решение проблемы различных словоформ [15] и опечаток [9].

5. К полученному вектору вхождения 4-грамм применяется схема взвешивания TF-IDF. Это стандартная нормализация в информационном поиске (описанная в [9]). Приведем краткое описание этой нормализации.

   Пусть рассматривается твит $d \in D$, где D — корпус твитов. Обозначим за $tf(w, d)$ количество вхождений 4-граммы w в твит d, а также рассмотрим величину $idf(w, D) = \log \frac{|D|}{|D_w|}$, где $D_w = \{d \in D : w \in d\}$ (рассматриваются только 4-граммы, входящие хотя бы в один из документов). Тогда вес TF-IDF этой 4-граммы определяется следующим образом: $TF\text{-}IDF(w, d, D) = tf(w, d)\, idf(w, D)$. Неформально, такая нормализация позволяет уменьшить вес 4-грамм, которые встречаются в большинстве документов корпуса, и увеличить вес 4-грамм, которые часто встречаются лишь в конкретном документе. В данном контексте эта схема позволяет снизить вес окончаний и стоп-слов для полученных 4-грамм.

6. Приведения полученного вектора к единичной l_2-норме. Опыт исследователей [16] показывает, что такая нормализация эффективна для линейных методов.

7. К полученному вектору признаков применяется логистическая регрессия.

Для предобработки и получения признаков использовалась библиотека машинного обучения scikit-learn [17], для классификации применялся пакет liblinear [12].

1.4 Метрики качества

По статистике, только 9% исходного потока сообщений из Твиттера, найденных по запросу про Яндекс и его продукты, выражают тональность по отношению к компании Яндекс. Это делает первый этап классификации задачей с несбалансированной выборкой, для которой многие исследователи ([18], [19]) предлагает использовать меры качества, отличные от доли верных ответов (accuracy). В частности, подходят меры точность P, полнота R и F-мера, которые и будут использоваться в дальнейшем в данной работе. Значения этих показателей измеряются для классов положительных и отрицательных сообщений (при этом применяется мера $F_{9/7}$, поскольку для данной задачи идеальным показателем можно считать $R = 90\%$ и $P = 70\%$), а также для класса тональных сообщений.

2 Предварительное исследование для определения узкого места по качеству

Предложенный способ классификации представляет собой конвейер (англ. “pipeline”). На первом уровне разделяются нейтральные и тональные сообщения, а на втором — положительные и отрицательные. При этом важными классами являются положительные и отрицательные сообщения. Стандартной техникой для улучшения такой системы является анализ верхних границ качества конвейера (англ. “pipeline ceiling analysis”) [20]. Суть этой техники следующая: слева направо нужно давать 100% качество для элементов конвейера и смотреть на выходные метрики. Потом рассчитать изменения качества между слоями и работать над слоем с наибольшим приростом. В данном случае такое исследование сводится к следующему: нужно выбрать класс только тональных сообщений и посмотреть, как алгоритм разделяет его. Далее в табл. 1 приводятся полученные результаты для теста tweets6000 при обучении на tweets26000.

Тип теста	$F_{9/7}^{tonal}$	$F_{9/7}^{positive/negative}$	R^{tonal}	$R^{positive}$	$R^{negative}$	P^{tonal}	$P^{positive}$	$P^{negative}$
Ideal	1	0,719	1	0,646	0,789	1	0,654	0,786
Baseline	0,441	0,345	0,901	0,464	0,808	0,239	0,282	0,158

Таблица 1: Сравнение качества иерархических классификаторов в зависимости от обычного/идеального первого уровня

Целевой метрикой изначально является $F_{9/7}^{positive/negative}$ (F-мера, усредненная по классам положительных и отрицательных сообщений). Если сделать идеальным первый уровень классификации, получим прирост $\delta_1 = 0,374$, а если сделать идеальным второй при наличии идеального первого, то будет прирост $\delta_2 = 0,281$. Таким образом, наибольший потенциальный прирост качества можно получить на первом слое.

Эти результаты говорят о том, что отделение положительных сообщений от отрицательных — простая задача, если изначально удалось хорошо разделить нейтральные и тональные. Поэтому основные усилия в данной работе сосредоточены именно на задаче отделения нейтральных от тональных.

Далее качество измеряется на выборке tweets26000 по схеме кросс-валидации по 10 фолдам. Более подробно, выборка $X = \{x_i\}_{i=1}^{L}$ случайным образом делится на 10 примерно равных непересекающихся частей (фолдов) $X_k, k = 1, \ldots, 10 : \bigsqcup_{k=1}^{10} X_k = X$. После этого обучение алгоритма классификации производится на 9 фолдах, которые соответствуют $X \setminus X_k$, а качество Q_k (это может быть P, R или F-мера) измеряется на одном оставшемся фолде X_k (будем называть его тестовым). В качестве тестового фолда последовательно выбирается каждый $X_k, k = 1, \ldots, 10$, а оценкой кросс-валидации является среднее качество: $Q_{CV} = \frac{1}{10} \sum_{k=1}^{10} Q_k$.

3 Сэмплирование Гиббса для несбалансированных выборок с дубликатами

Известно, что для работы с несбалансированными выборками должны применяться специальные методы. Это связано с теми функционалами качества, которые оптимизируют алгоритмы машинного обучения. Такие функционалы обычно являются верхней оценкой для доли верных ответов (accuracy), поэтому и оптимизируется в некотором смысле именно эта величина. Однако, целевой метрикой в задаче отделения тональных от нейтральных является F-мера либо точность P при выбранном уровне полноты R.

3.1 Метод бутстрэпа

Одним из способов учета несбалансированности выборки является итерационный сбор сбалансированной обучающей выборки (который обычно называют bootstrap). Идея этого метода заключается в следующем:

1. На первой итерации берутся все объекты из класса-меньшинства и такое же количество объектов из класса-большинства выбирается случайно. На полученных данных обучается классификатор.

2. На следующей итерации полученный классификатор применятся ко всем объектам из класса-большинства, для каждого объекта получаем вещественную оценку — вероятность принадлежать классу-меньшинству.

3. Объекты сортируются по этой оценке, выбираются объекты с наибольшими оценками (и, как следствие, наибольшей вероятностью ошибки). Количество объектов берется равным количеству объектов класса-меньшинства.

4. На полученной выборке обучается вторая итерация классификатора.

5. И так далее.

Мотивация использования такой процедуры заключается в том, чтобы отобрать для обучения алгоритма самые сложные примеры из класса-большинства.

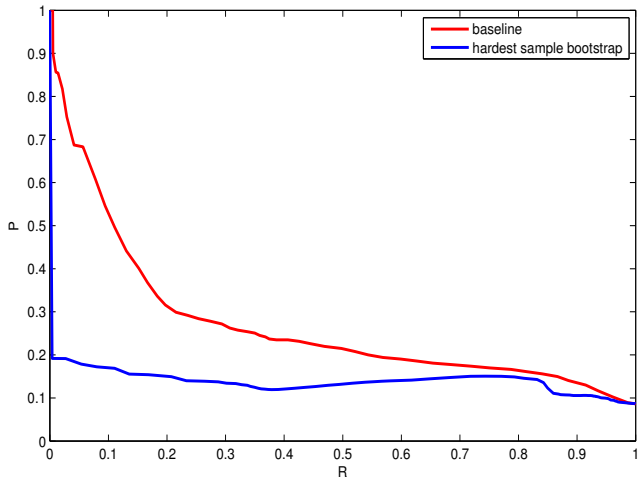

Рис. 1: Зависимость точности по классу тональных от полноты, сравнение бутстрэпа (hardest sample bootstrap) и эталонного метода (baseline)

3.2 Качество бутстрэпа в задаче анализа тональности

К сожалению, прямое применение идеи бутстрэпа к выборке тональный/нейтральный приводит к очень низкому качеству. Кривая полноты/точности для класса тональных сообщений приводится на рис. 1.

По графикам видно, что бустрэп только ухудшил качество, причем значительно. Поскольку в выборке присутствуют полудубли (почти совпадающие сообщения), самые сложные отобранные примеры могли оказаться недостаточно разнообразными. Для увеличения разнообразия в данной работе предлагается использовать вероятностное сэмплирование для порождения выборки.

3.3 Вероятностный бутстрэп

Пусть для каждого объекта x_i нам известна вероятность неправильной классификации p_i^{error}. Введем на объектах распределение $P_i \sim \exp^{-\frac{1-p_i^{error}}{T}}$. Здесь T — параметр температуры, который отвечает за "остроту" пиков распределения. Тогда можно случайным образом породить необходимое число примеров из вышеописанного мультиномиального распределения. Перебором по сетке было найдено оптимальное значение T=3.

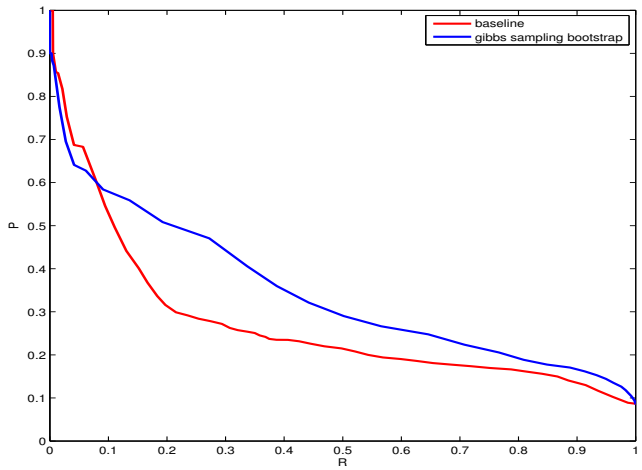

Рис. 2: Зависимость точности по классу тональных от полноты, сравнение вероятностного bootstrap и baseline

Полученные результаты для такого вероятностного сэмплирования приведены на рис. 2.

Видно, что предложенный метод успешно решает проблему недостаточного разнообразия и, кроме того, позволять повысить качество по сравнению с обучением на несбалансированной выборке в эталонном алгоритме.

4 Учет синтаксиса, морфологии и эмоциональных меток слов в задаче анализа тональности

4.1 Идеи улучшения признаков

Использование 4-грамм в эталонном алгоритме помогает решить проблему морфологического разнообразия слов в русском языке, увеличивая статистику по входящим в твит словам. Однако, такой подход приводит к появлению ложных тональных слов. Например, слово “хоровод” при таком подходе даст значительный положительный вклад, поскольку первая 4-грамма “хоро” также встречается в тональном слове “хороший”. Более точный подход связан с использованием лемматизации, при которой для слова находится его корректная начальная форма. Далее используются только лемматизированные словоформы.

Также важно понимать ограничения модели униграмм, которая была используется в эталонном алгоритме. Она не учитывает взаимное расположение слов и синтаксические связи между ними. Это плохо тем, что тональные слова в твите могут относиться не к объекту оценки, а к другому слову, при этом алгоритм не будет знать об этом. Приведем несколько реальных примеров:

- “вроде для яндекс.фотки тоже плагин есть для айфото. Фликр неудобный помоему”. Слово “неудобный” несет тональность, эта тональность отрицательная. Однако, она никак не относится к сервису Яндекс.Фотки.

- “яндекс пробки показывает оптимистичную картину”. Непонятно, является ли это сарказмом, но к сервису Яндекс.Пробки тональность никак не выражается. При этом есть тональное слово “оптимистичный”.

- “Посоветуйте хороший торрент клиент для Ubuntu и нормальный аналог Пунто Свитчера”. Тональное слово “хороший” никак не относится к продукту Яндекса “Punto Switcher”.

Для устранения этого недостатка будут рассмотрены дополнительные признаки, которые будут описаны в следующих подразделах. Далее описываются структуры, на основе которых строятся признаки, и сам метод порождения признаков.

4.2 Биграммы и трехграммы по последовательности

Предлагается использовать словесные подстроки s длиной два и три слова, называемые биграммами и трехграммами соответственно. Например, если есть фраза “Яндекс мне очень помог”, то в ней будут содержаться:

- Биграммы: “Яндекс мне”, “мне очень”, “очень помог”.
- Трехграммы: “Яндекс мне очень”, “мне очень помог”.

Трехграммы являются более точными индикаторами наличия тональности, однако статистики по ним меньше. Например, если встретилась трехграмма “мне очень помог”, то количество ее вхождений в размеченный корпус твитов будет невелико (возможно, это будет единственный твит, в котором пользователь выразил свое отношение именно такими словами). Отсюда следует, что нужен либо очень большой корпус для обучения (которого, как уже было отмечено, нет для задачи тональный/нейтральный), либо полезность трехграмм будет весьма ограниченной. Более подробно, если трехграмма является редкой:

- Она повлияет на классификацию малого количества документов.
- Ее вхождение в документы только одного класса не позволит сделать вывод о том, характерна ли она для этого класса. Она может встретиться случайно и не относиться к объекту оценки, например.

Именно по указанным выше причинам в задачах анализа естественных языков не используются 4-граммы по словам: статистики по ним будем совсем мало. Для биграмм же статистики больше, поэтому их полезность выше, чем для трехграмм. При этом они все-таки учитывают взаимное расположение слов, в отличие от униграмм.

4.3 Биграммы и трехграммы по синтаксическому дереву разбора

Для синтаксического дерева нет стандартных процедур порождения биграмм и трехграмм, поэтому предлагается ввести кодировку на основе ребер дерева. Пусть

есть фраза “Яндекс мне помог”. Тогда в дереве будут ребра “помог → Яндекс”, “помог → мне”. Понятно, что первое ребро и конструкция в целом полезна для распознавания тональности, выраженной по отношению к объекту. Предлагается учитывать ребра и два типа троек: тройки {x → y, x → z} и тройки {x → y, y → z}. Такие ребра и тройки кодируют локальную структуру дерева, так же как обычные биграммы и трехграммы кодируют локальную структуру последовательности. Кроме того, синтаксическое дерево позволяет “подклеить” отрицательную частицу “не” к тем словам, к которым эта частица относится. Например, “не помог” преобразуется в “не_помог”.

4.4 Морфологические признаки и метки эмоциональности

Дополнительная морфологическая информация, которую тоже следует учесть, включает метки частей речи, которые являются одним из результатов работы лемматизатора. Если от объекта оценки (Яндекс) есть синтаксическая связь с каким-либо прилагательным, это может быть косвенным признаком выражения тональности. Среди меток частей речи следует ввести дополнительные для эмотиконов (например, “:)”) и восклицаний (например, “!!!”), потому что они тоже могут быть выражениями тональности.

Еще одной полезной идеей является использование меток эмоциональности слова. В данном эксперименте используются метки, полученные на основе ручного сбора порядка 2000 тональных слов. Для положительных слов была проставлена метка “positive”, для отрицательных — “negative”. В метке эмоциональности также учитывается наличие отрицания за счет использования синтаксического дерева разбора: при наличии отрицательной частицы тональность изменяется на противоположную.

4.5 Метод порождения признаков

В данном разделе будет описан метод порождения признаков на основе структур, описанных в предыдущих разделах.

Пусть задано множество графов с занумерованными вершинами $S_v = \{G_v^i\}_{i=1}^K$, где $G_v = (V, E, N)$, $N : \{1, \ldots, |V|\} \to V$ — обратное нумерующее биективное отображение, $E \subset \{1, \ldots, |V|\} \times \{1, \ldots, |V|\}$. Здесь имеется в виду, что параметром множества и графа является тип вершины v. Нас будут интересовать два случая:

1. Вершиной v_t является множество $\{l, p, t, w\}$, где $l \in L$ — лемма слова, $p \in P$ — метка части речи, $t \in T$ — метка тональности слова, w — специальный элемент (смысл в том, чтобы признаки группировались по нему; такой элемент в англоязычной литературе обычно называют “wildcard”).

2. Вершиной v_e является один из элементов множества $L \cup P \cup T \cup \{w\}$.

Введем кодирующее отображение на графах:

$$F : S_{v_t} \to 2^{S_{v_e}},$$

$$G_t = (V_t, E_t, N_t),$$

$$G_e = (V_e, E_e, N_e),$$

$$F(G_v) = \{G_e \in S_e : |V_e| = |V_t| = N,\ E_e = E_t,\ N_e(i) \in N_t(i),\ i = 1, \ldots, N\}.$$

По сути, это отображение задает декартово произведение на графе множеств.

Теперь остается ввести признаки классификации. Пусть дан твит $d \in D$. По нему можно построить униграммы, биграммы и трехграммы по последовательности и синтаксическому дереву. Все эти структуры, описанные в предыдущих разделах, являются графами G_t. Пусть извлечение множества структур-графов из твита d задается отображением $U : d \to \{G_t^i\}_{i=1}^{K_d}$. Тогда каждого твита d признаками классификации будут tf (то есть число вхождений в твит) униграмм, полученных отображением $F(U(d))$.

4.6 Результаты экспериментов

Предложенные признаки позволяют заметно улучшить качество по сравнению с комбинацией “эталонный алгоритм”+“вероятностный бутстрэп”. Кривая полноты/точности для полученного классификатора приводится на рис. 3.

Интересно отметить, что вероятностный бутстрэп еще более важен для новых признаков, чем для простых униграмм. Это следует из рис. 4, на котором можно наблюдать еще более значительную разницу, чем ранее для униграмм.

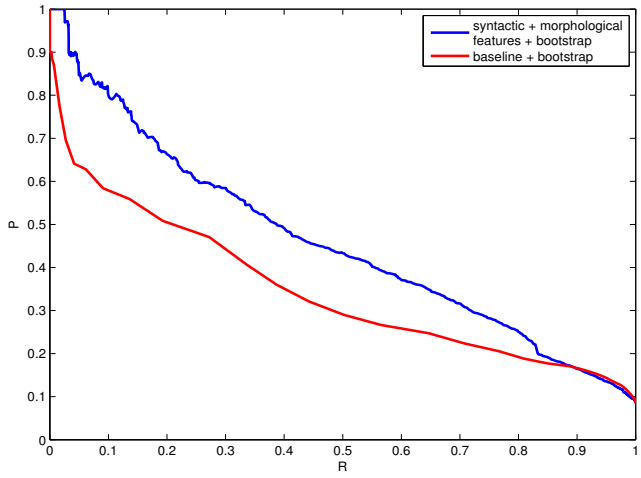

Рис. 3: Зависимость точности по классу тональных от полноты, сравнение синтаксических и морфологических признаков с эталоном (в обоих случаях используется вероятностный бутстрэп)

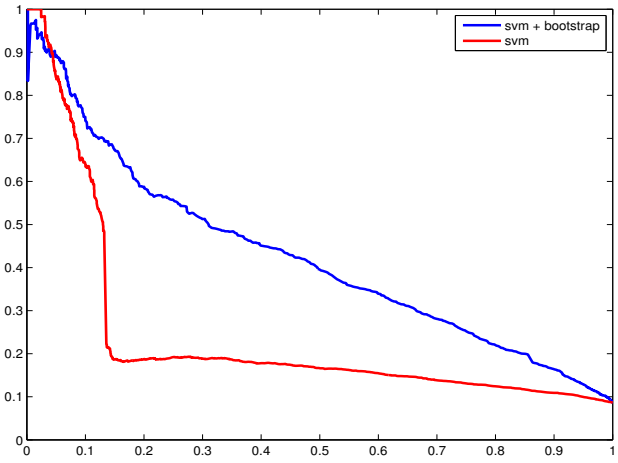

Рис. 4: Зависимость точности по классу тональных от полноты, сравнение с/без бутстрэп при использовании синтаксических и морфологических признаков

5 Сравнение эффективности логистической регрессии и градиентного бустинга

5.1 Мотивация

Мотивацией к использованию градиентного бустинга являются результаты соревнования Job Salary Prediction на известном ресурсе по машинному обучению Kaggle [21]. Задача этого соревнования ставилась как регрессия по текстам. Команда автора данной работы "natural_intelligence" заняла 15-ое место среди 294 команд и использовала градиентный бустинг и отсечение по документной частоте для униграммных признаков (будет описано далее). При этом линейная регрессия из известного пакета Vowpal Wabbit, использованная командой "Keyser Söze", не поднялась выше 36-ого места. Эксперименты автора данной работы также показали низкую эффективность линейных методов в задаче из соревнования.

Поскольку градиентный бустинг существенно обошел линейные методы в похожей задаче по текстам, его использование кажется разумным и в данной задаче.

5.2 Эксперимент

Поскольку бустинг не может работать с таким большим количеством признаков, которое получается при указанном методе порождения (их порядка 600000), в первую очередь необходимо отобрать наиболее полезные (напомним, что в предложенной схеме все признаки представляют собой униграммную кодировку структур, содержащих слова). Одним из способов провести такой отбор является сортировка признаков по количеству самплов, в которых признак-униграмма присутствует, и отбрасывание признаков с низкой частотой в корпусе. В соответствии с законом Ципфа, предполагается падение встречаемости униграмм по закону $N \sim 1/x^{\gamma}$, где x — номер униграммы в отсортированном по убыванию частоты списке, $\gamma > 0$ — константа (на практике обычно порядка 1), N — количество вхождений униграммы в корпус. Таким образом, отброшенные униграммы влияют на малое количество документов и не так важны для распознавания. Кроме того, если признак встречается в малом числе документов, статистики по нему недостаточно для использования в распознавании.

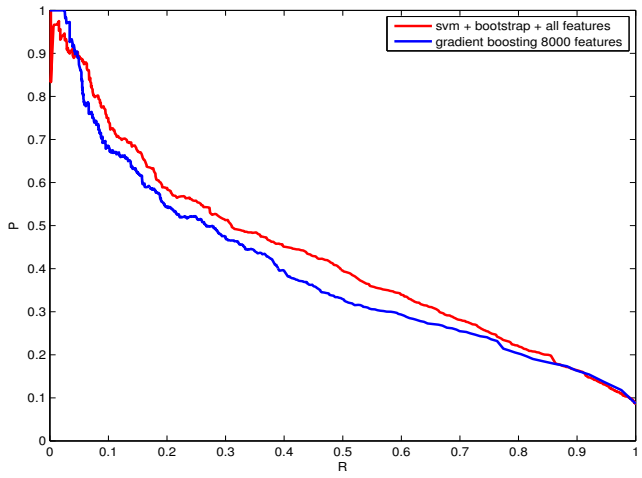

Рис. 5: Зависимость точности по классу тональных от полноты, сравнение градиентного бустинга и логистической регрессии

Применение этой идеи позволило сократить признаковое описание до 8000 (применялось условие $N \geq 10$) и использовать бустинг. Использовался градиентный бустинг [22] из пакета машинного обучения scikit-learn [23]. Следует отметить, что несбалансированность выборки не так важна для бустинга, как для логистической регрессии. Поэтому вероятностный бутстрэп не используется для бустинга, а для логистической регрессии используется. Из полученной кривой полноты/точности, приведенной на рис. 5, следует, что градиентный бустинг оказывается чуть хуже логистической регрессии.

6 Учет мета-информации из Твиттера

6.1 Идеи улучшения признаков

Ранее рассматривалась только текстовая информация, содержащаяся в сообщении, то есть слова и лингвистические связи между ними. Однако, кроме этой информации есть еще данные про взаимосвязи сообщений, данные про пользователей, скрытая во внешних http-ссылках информация, информация о найденных ключевых словах в поиске по блогам. Предлагается использовать следующие данные в качестве признаков:

1. Имя пользователя. Использование этих данных важно, поскольку значительную часть нейтральных сообщений составляют сообщения из новостных потоков, генерируемых газетами и компаниями. Эти потоки являются пользователями, поэтому информация в имени позволит лучше фильтровать сообщения.

2. Наличие ретвита и наличие непустого ретвита (под ретвитами в данном случае понимаются еще и ответы пользователей: между этими понятиями нет принципиальной разницы в рамках данной задачи). Для этого выполняется деление цепочки ретвитов на отдельные сообщения. Анализ содержания ретвита полезен, поскольку непустой ретвит с большей вероятностью несет тональность, а для пустого возможны разные варианты. Кроме того, сам факт ретвита может быть связан с наличием тональности.

3. Является ли внешняя ссылка файлом или путем (заканчивается на знак "/" или на домене).

4. Наличие домена yandex во внешней ссылке. Это может быть косвенным признаком нейтральности сообщения, которое появилось в выдаче ППБ только из-за домена в ссылке. Однако, на этапе информационного поиска такие сообщения отбрасывать нельзя, потому что ссылка тоже может быть объектом оценки.

5. Ключевые слова, выделенные тэгом <b> в выдаче поиска по блогам. Таким образом сохраняется информация о том, почему поиск счел данное сообщение релевантным запросу (можно считать эти слова объектом оценки).

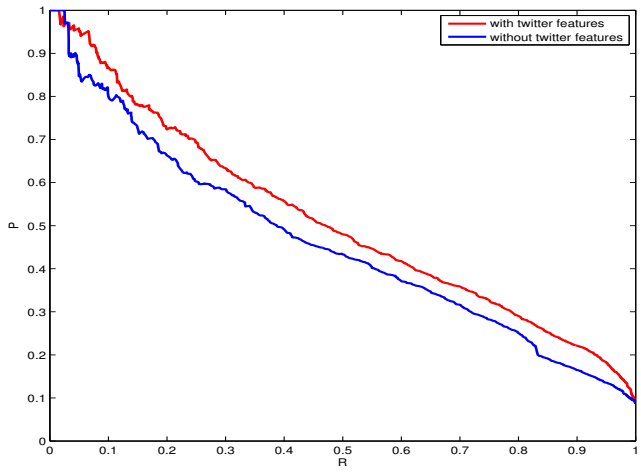

Рис. 6: Зависимость точности по классу тональных от полноты, сравнение без учета/с учетом мета-информации Твиттера

Чтобы из приведенных данных получить признаки, достаточно закодировать их словами по схеме “префикс” + “номинальный признак” и по полученному множеству униграмм рассчитать tf-признаки. Например, если нет ретвита, можно взять униграмму “no_retweet”, а если встретилось ключевое слово “яндекс” — то “key_word_яндекс”.

6.2 Эксперимент

Для проведения эксперимента фиксировалась комбинация “алгоритм логистической регрессии”+“вероятностный бутстрэп”+“синтаксические признаки”, описанный в разделе 4. В этот алгоритм добавлялись описанные ранее мета-признаки. Полученная кривая полноты/точности приведена на рис. 6. Из сравнения кривых можно сделать вывод, что учет мета-информации значительно улучшает качество распознавания, особенно в зоне высокой полноты. Следует отметить, что именно эта зона представляет практический прикладной интерес, поэтому рост качества в ней особенно важен.

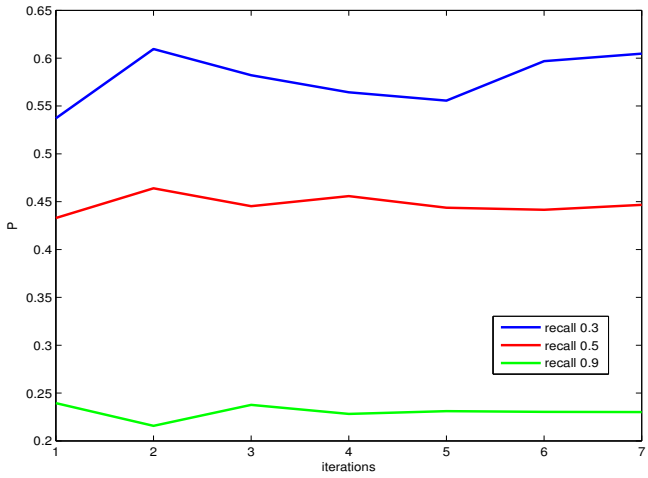

Рис. 7: Зависимость точности по классу тональных от номера итерации вероятностного бутстрэпа

7 Зависимость качества алгоритма от параметров

7.1 Качество вероятностного бутстрэпа в зависимости от числа итераций

Ранее число итераций бутстрэпа было зафиксировано значением 2 ввиду отсутствия дальшейшего роста качества. В данном разделе этот вопрос исследуется более подробно и приводятся графики качества. Под качеством понимается точность P при фиксированном уровне полноты R. Графики соответствуют нескольким уровням полноты R: 0.3, 0.5, 0.9. Они приведены на рис. 7.

Из графиков можно сделать два вывода:

1. Оптимальное значение числа итераций равно 2. Для большего числа итераций заметны колебания качества, которые постепенно сходятся к некоторому уровню насыщения, более низкому по сравнению со значением качества для двух итераций.

2. Метод вероятностного бутстрэпа эффективен в области низкой (0.3) и средней (0.5) полноты. С увеличением полноты эффективность этого метода снижается. В области высокой полноты (0.9) его применение приводит к ухудшению качества по сравнению со случайной балансировкой данных.

7.2 Сравнение методов работы с несбалансированными выборками в области высокой полноты

Кроме бутстрэпа существует еще один известный метод работы с несбалансированными выборками — назначение классу-меньшинству большего веса. Таким образом, штраф за ошибку на этом классе становится выше. В данном разделе исследуется, какой метод оказывается лучше по метрике “точность на уровне полноты 90%”. Далее эта метрика обозначается P_{90}. Следует еще раз отметить, что именно область высокой полноты является практически интересной в задаче промышленного анализа тональности. Поэтому указанная метрика представляет интерес с точки зрения практической применимости метода.

Для исследования качества обоих методов были введены сетки по параметрам:

1. Для вероятностного бутстрэпа перебирались значения параметра $\frac{1}{C}$ (C — параметр регуляризации логистической регрессии) в диапазоне $[0.001, 1]$, а также значения температуры в диапазоне $[0.01, 20]$. По каждому параметру сетка содержит 10 значений, то есть всего 100 точек.

2. Для метода с весами перебирались значения параметра $\frac{1}{C}$ (C — параметр регуляризации логистической регрессии) в диапазоне $[0.001, 0.5]$, а также значения веса объектов класса тональных сообщений w в диапазоне $[1, 60]$. По каждому параметру сетка содержит 10 значений.

По итогам максимизации качества по сетке (лучшая точка по сетке являлась стартовой для метода оптимизации Нелдера-Мида) было установлено, что вероятностный бутстрэп позволять достичь качества $P_{90} = 0.25$, в то время как метод с весами — $P_{90} = 0.2625$. Таким образом, в области высокой полноты метод с весами оказывается более эффективен. Далее производится исследование этого метода и приводятся значения оптимальных параметров для него.

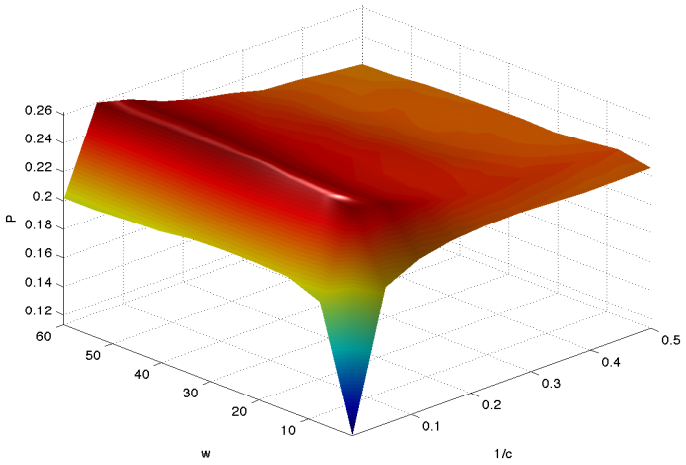

Рис. 8: Зависимость от параметров точности P_{90} по классу тональных

7.3 Зависимость качества от параметров для алгоритма с весами

У предложенной схемы есть два параметра: параметр регуляризации логистической регрессии $\frac{1}{C}$ и вес w объектов класса-меньшинства. Двухмерная поверхность зависимости качества от указанных параметров (построенная по сетке из предыдущего раздела) приводится на рис. 8. Оптимальными параметрами являются $\frac{1}{C} = 0.049$ и $w = 27.90$. Стоит отметить, что оптимальным значением веса w является не 10.49, которое соответствует выравниванию сумм весов тональных и нейтральных сообщений в обучающем множестве (именно во столько раз тональных больше, чем нейтральных). Однако, именно в районе $w = 10$ наблюдается резкий скачок качества, а при дальнейшем увеличении веса качество растет незначительно. Таким образом, в целом идея выравнивания сумм весов оказывается адекватной выбранной метрике качества P_{90}.

7.4 Зависимость качества от размера выборки

Для размера выборки на рис. 9 приводится график качества при зафиксированных оптимальными значениями двух других параметрах. Заметно переобучение: зазор между качеством на обучении и контроле остается значительным даже для мак-

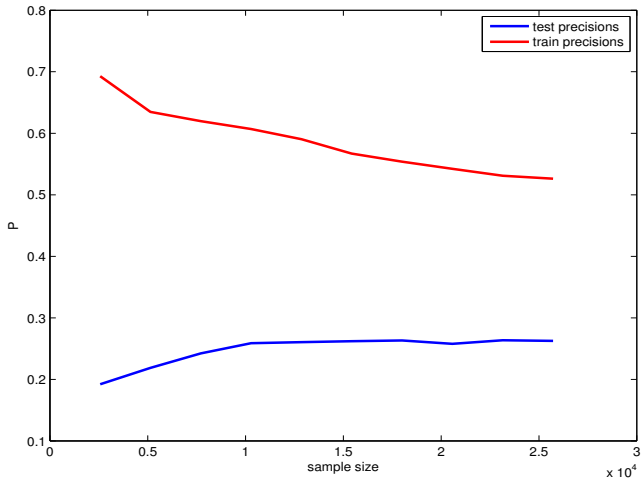

Рис. 9: Зависимость точности на обучении (train) и контроле (test) от размера выборки

симального размера выборки. Однако, эксперименты показывают, что уменьшение переобучения в данной задаче не приводит к росту качества: обе кривые опускаются и становятся ближе друг к другу.

8 Обсуждения и выводы

По итогам обзора литературы и проведенного исследования было установлено, что задача определения тональности сообщения в Твиттере относительно компании является сложной для машинного обучения. Классический униграммный подход работает плохо, и необходимо использовать дополнительную информацию — лингвистические признаки и мета-данные из Твиттера.

Несмотря на значительное улучшение качества по сравнению с эталоном, основными проблемами предложенного метода остаются:

1. Недостаточный размер выборки для подзадачи нейтральный/тональный. Даже при максимальном размере выборки 26000, среди них только 2600 сообщений являются тональными. Интуитивно понятно, что такого количества недостаточно для решения данной задачи (естественный язык обладает большим разнообразием форм выражения эмоций). Хотя на рис. 9 наблюдается лишь слабый рост качества с ростом размера выборки, статистики явно недостаточно, чтобы говорить о насыщении. В отличие от классических "кривых обучения", у данной кривой множество признаков не фиксировано и увеличивается с размером выборки. Это дает основания полагать, что при увеличении размера выборки качество продолжит расти.

2. Отсутствие масштабной разметки слов по тональности. В данной работе применяется словарь размером примерно 2000 слов, и этого явно недостаточно. Реальные примеры:

   - "Я смотрю Yandex Fotki сильно допилили за год. Думаю, может туда переехать с Flickr. Хм?". В данном сообщении пользователь выражает положительное отношение к сервису Яндекс.Фотки. Однако, алгоритм присвоил ему лишь 33% вероятность быть тональным. Проблема в том, что словосочетание "сильно допилили" не учтено в словаре тональных слов.
   - " Моё доверие яндекс утратил...Как так можно...Перепутать Россия 2 и Первый? Когда Россия-Канада играют... ". Здесь выражается отрицательная тональность, но алгоритм присваивает лишь 55%

вероятность быть тональным. Проблема, опять же, со словом “утратил”, которого нет в словаре тональных.

3. Недостаточно качественное выделение объекта оценки. Пример реального сообщения: “закончится эта неделя, закончится мой круг ада, тобеш закончатся все контрольные и зачеты”. Здесь Поиск по Блогам выделил “мой круг” из-за сходства с сервисом Яндекса “Мой Круг”. Однако, в данном контексте эта фраза несет другой смысл. При этом из-за наличия сильных тональных слов этот твит был отнесен к тональным с вероятностью 82%.

4. Сложность машинного анализа смысла естественного языка. Пример: “не нравится мне, что в яндекс-картах весь литейный красный;(((”. Это сообщение является тональным с вероятностью 90% по оценке алгоритма. Однако, только человеческие знания о том, что “литейный” не является атрибутом Яндекса, помогают отнести его к нейтральным.

9 Заключение

В данной работе получены следующие результаты:

- Разработан 2-х этапный метод классификации эмоциональной окраски сообщений.
- Предложен метод вероятностного бутстрэпа для несбалансированных выборок с полудублями.
- Предложен метод учета дополнительной информации о морфологии, синтаксисе, метках тональности слов и мета-информации из Твиттера.
- Показано, что в совокупности применение предложенных методов позволяет улучшить точность и полноту.

Список литературы

[1] Shima Gerani, Mark James Carman, and Fabio Crestani. Proximity-based opinion retrieval. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '10, pages 403–410, New York, NY, USA, 2010. ACM.

[2] http://blogs.yandex.ru/.

[3] Bing Liu. Sentiment analysis tutorial. *Talk given at AAAI-2011, Monday*, 2011.

[4] http://nlpseminar.ru/archive/lecture44/.

[5] Dan Jurafsky and Christopher Manning. https://class.coursera.org/nlp/lecture/preview, sentiment analysis lecture.

[6] Luciano Barbosa and Junlan Feng. Robust sentiment detection on twitter from biased and noisy data. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, COLING '10, pages 36–44, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[7] Xiaolong Wang, Furu Wei, Xiaohua Liu, Ming Zhou, and Ming Zhang. Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, CIKM '11, pages 1031–1040, New York, NY, USA, 2011. ACM.

[8] http://help.yandex.ru/search/?id=1111313.

[9] Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, 2008.

[10] Minmin Chen, Zhixiang Xu, Kilian Weinberger, and Fei Sha. Marginalized denoising autoencoders for domain adaptation. 2012.

[11] Bo Pang and Lillian Lee. A sentimental education: sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the*

42nd Annual Meeting on Association for Computational Linguistics, ACL '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[12] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[13] http://hunch.net/ vw/.

[14] http://hunch.net/?p=2094.

[15] James Mayfield and Paul McNamee. Single n-gram stemming. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, SIGIR '03, pages 415–416, New York, NY, USA, 2003. ACM.

[16] http://www.csie.ntu.edu.tw/ cjlin/liblinear/faq.html.

[17] http://scikit-learn.org/dev/modules/feature_extraction.html.

[18] Andrew Ng. https://class.coursera.org/ml/lecture/preview, advice for applying machine learning.

[19] Dan Jurafsky and Christopher Manning. https://class.coursera.org/nlp/lecture/preview, text classification lecture.

[20] Andrew Ng. https://class.coursera.org/ml/lecture/preview, application example (photo ocr).

[21] https://www.kaggle.com/c/job-salary-prediction.

[22] http://scikit-learn.org/dev/modules/generated/sklearn.ensemble.gradientboostingclassifier.html.

[23] http://scikit-learn.org/.