

**(а) задача, которую решали авторы, какие преимущества, по мнению авторов, их подход дает**

Авторы статьи решали задачу автоматической генерации домен-специфичного русского тонального словаря. Преимущества подхода, по мнению авторов, состоят в том, что:

1. Для работы их алгоритма не требуется никакая ручная разметка, а только довольно крупный корпус, который не так трудно собрать.
2. Нет привязки к конкретному языку: авторы работали с русским, но можно использовать и в других языках.
3. Алгоритм применим для любого домена текстов, и предназначен для выделения домен-специфичных слов. Таким образом, для переноса на другой домен достаточно лишь изменить корпус.

**(б) с какими другими подходами они себя сравнивают (кратко);**

Авторы кратко пересказывают, какие другие подходы используются для выделения тональной лексики:

- подход, основанный на обучении нескольких классификаторов на одном домене, а затем обобщении результатов на другие домены (Четверкин, Лукашевич 2012);
- подход, использующий собранные вручную словари на английском и испанском, для автоматической генерации тональных словарей на других языках (немецком, русском, итальянском и других). (Steinberger J., Ebrahim M., Ehrmann M., Hurriyetoglu A., Kabadjov M. A., Lenkova P., Steinberger R., Tanev H., Vazquez S., Zavarella V. 2012);

Затем рассматривается, как в сентимент-анализе применяются графы

- для переноса тонального лексикона, выделенного в размеченном документе, на неразмеченный корпус (Wu Q., Tan S., Cheng X. 2009),
- для представления предложений и отношений между ними в задаче выделения тональных предложений (Pang B., Lee L. 2004),
- в анализе тонального лексикона социальных сетей некоторые подходы строят граф на основе WordNet, а затем случайно обходят его, чтобы ранжировать полярность тональных слов (Esuli A., Sebastiani F. 2007).

**(в) какой датасет, объем, предварительная разметка; какие другие источники они использовали**

Для обучения авторы использовали 259 023 деперсонализированных неразмеченных рецензии на отели, которые были размещены в интернете. В статье не указано, из каких конкретно ресурсов были взяты эти тексты. По словам авторов, размер датасета — 660 МБ. (Не очень понятно, для чего эта информация без указания кодировки, которую авторы использовали, и формата, в котором хранились данные, — количество токенов в корпусе было бы более информативным.) Изначально датасет никак не

размечен — нет лейблов, которые бы как-то кодировали общую оценку отзыва по аспектам или без них.

### **(г) что служило признаками при обучении**

В данном случае, модель - это граф, а признаки в модели - это прилагательные и то, сколько раз они встретились друг с другом в разных союзных конструкциях (сочетания и противопоставления, которым соответствуют веса на рёбрах). Следует заметить, что привычные нам модели машинного обучения (SVM, RandomForest и т. п.) здесь не используются, поэтому едва ли можно говорить о признаках (features) для обучения — обучения как такового нет.

### **(д) алгоритм: в чем "фишка" технологии**

Алгоритм, который предлагают авторы статьи заключается в использовании сочинительных союзов, которые соединяют прилагательные и наречия, и противительных союзов, которые разделяют их. В сочинительной конструкции слова будут одной полярности, а в противительной — противоположной: “вкусный и сытный”, “вкусный, но дорогой” и т. д.. Это позволяет построить взвешенный граф, где узлы - прилагательные и наречия, а ребра — связи между ними, вес которых будет числом предложений, где слова связаны между собой сочинением или противопоставлением. Кроме того, сочинение с отрицанием считалось противопоставлением (“вкусно и не очень дорого”), а противопоставление с отрицанием считалось сочинением (“вкусно, но не очень дорого”)

Первый этап алгоритма — построение графа. Слова считались связанными, если удовлетворяли такому шаблону: (ADV | NEG) \* ADJ(, ? (AND | BUT)? (ADV | NEG) \* ADJ) +, — и были в одном роде, числе и падеже. Для разметки тренировочного датасета (морфологического парсинга) использовался Mystem. Наличие отрицания учитывалось при построении связей. Авторы попробовали два подхода для работы с прилагательными с приставкой не-: в первом они считали слова типа “большой” и “небольшой” разными словами, а во втором — одним словом, но учитывали не- как отдельно стоящее отрицание.

Тренировочный корпус состоял из отзывов интернет-пользователей, поэтому авторам пришлось учитывать возможные грамматические и пунктуационные ошибки.

Как взвешивали ребра:  $\text{weight}(w1, w2) = n(w1 \text{ AND } w2) - K * n(w1 \text{ BUT } w2)$ , где K - это коэффициент, который нужен, чтобы компенсировать недостаток негативных связей.

Далее авторы пишут о том, как они определяли расстояние между “полюсами” — множествами тональных слов одной полярности — и узлом-кандидатом, здесь у них тоже два варианта: считать этим расстоянием самую тяжелую связь между словом и всеми словами одной полярности или же сумму весов всех связей со всеми словами этой полярности.

Так как для recall требуется извлечь все тональные слова, чтобы проверить, сколько из них нашел алгоритм, в обработать датасет такого объема вручную невозможно,

авторы выбрали случайно 500 статей, а затем разметили руками все прилагательные в них на два класса: позитивные и отрицательные.

Для оценки precision, они взяли все слова, которые извлек их алгоритм, и разметили их на три класса: позитивные, отрицательные и нейтральные. Но алгоритм выделял только негативные и положительные оценки, нейтральные он не рассматривал, поэтому для оценки precision они исключали нейтральные из подсчета.

Для оценки качества алгоритма с разными значениями K, авторы считали F-меру.

### **(е) анализ таблички результатов**

1. Исключение нейтральных слов сильно повышает точность и полноту. (Не очень понятно, насколько это реально что-то показывает с учетом того, что алгоритм нейтральные не извлекает, соответственно, исключение нейтральных слов не делает, оно делается руками).
2. Если нейтральные слова отфильтрованы, то удаление “не-” из прилагательных значительно поднимает точность (на графиках это не очень заметно: явно повышает точность фильтрация нейтральных слов, а вот по графикам без нейтральных и с разными подходами к “не-” разницы особой не видно).
3. Если не удалять “не-”, то лучшие результаты F-меры получаются с  $K < 3$ .
4. Если удалять “не-”, но оставлять нейтральные слова, то лучше всего работает  $K=1$ . Если и нейтральные убрать, то с увеличением K от 1 до 10 F-мера растет.
5. При оптимальных параметрах (перечисленных выше) алгоритм выдает 79,9% точности и 75,4% полноты.

### **(ё) резюме по тому, как в статье анализируются ошибки (если анализируются)**

Ошибки, которые делает алгоритм, в статье не обсуждаются никак, хотя было бы интересно посмотреть, куда утекают 20 процентов точности и почти 30 процентов полноты.

### **ж) что вам кажется наиболее интересным (обещающим) в примененной технологии, что не нравится/кажется странным**

Главное преимущество подхода — никакой ручной разметки и никаких словарей-тезаурусов, кроме начальных сидов. Это позволяет использовать алгоритм на корпусах любого масштаба.

С другой стороны, для оценки качества алгоритма понадобилось разметить вручную достаточно значительную часть корпуса, а результаты недостаточно хороши, чтобы просто применять их дальше без проверки качества.

Здорово и интересно, что авторы попробовали разные подходы к работе с отрицательной приставкой у прилагательных, и проверили, как алгоритм работает с

коэффициентами от 1 до 10, но не хватает анализа: почему в одних случаях высокий K работает хорошо, а в других нет?

Довольно мало описан тренировочный корпус: откуда взяты тексты, как был проводился crawl, сколько токенов / уникальных токенов / лемм / прилагательных было в корпусе. Особенно полезно было бы знать тотальное число прилагательных, поскольку именно их ищет этот алгоритм.

Странно, что авторы анонсировали два подхода к нахождению расстояния до полюсов, но никаких результатов/графиков/выводов по этому поводу не предоставили.

В работе используется русский язык, но сиды перечислены только по-английски — это неудобно, потому что у некоторых слов может быть несколько переводов на русский язык. То же самое можно сказать и про союзы.

Плохо, что в подходе не учитываются наречия меры и степени при взвешивании ребер: кажется, что они могли бы повлиять на качество, причем авторы их вообще-то вытаскивают при поиске кандидатов.

Не очень хорошо, что не указано, как именно обрабатывались ошибки правописания. Они как-то задавались на уровне паттерна, но как конкретно — мы не знаем.

Еще один недостаток: кандидаты извлекаются “в лоб” — по паттерну. Хотелось бы проверить, не улучшится ли качество, если использовать синтаксические парсеры и вытаскивать синтаксические группы.

### **(3) что совсем непонятно (термины, методы и т.п.)**

Кажется, что такой метод принципиально не способен понять, что слово нейтральное, и не ранжирует слова на более и менее тональные. Кажется, можно было бы ранжировать исходя из информации в графе (веса на рёбрах), например, по расстоянию до центроида набора слов одной тональности.

Самый непонятный раздел статьи — это интерпретация результатов. Авторы указывают, что для оптимальной работы алгоритма нужно как-то исключать нейтральные слова, но их алгоритм такого не делает — нейтральные слова не выделяются. Пока что графики с исключением и без исключения нейтральных слов выглядят одинаково неинформативными, так как их различие — это ручная разметка проверочного датасета. (То есть исключение нейтральных слов

Не ясно, почему для precision руками размечали только на позитивные и негативные, а для recall еще нейтральные выделили. Видимо, это связано с тем, какие ошибки алгоритм делает, но обсуждения ошибок и обоснования этому никакого нет.