1. Приветствие
2. Данная работа представляет собой подход организации различных аспектов продукта, относящегося к категории электронные устройства, в иерархию на основе знаний о пользовательских отзывах. Основываясь на произвольной иерархии (построенной вручную), создается иерархическая организация отзывов потребителей по различным аспектам продукта и совокупным мнениям потребителей по этим аспектам. При такой организации пользователь может получить обзор потребительских мнений в максимально короткий срок.
3. Для того, чтобы при рассказе не было недопонимания, я вынесла несколько определений на слайд. Важно понять разницу между аспектом и идеальным аспектом, т.к. в дальнейшем я буду использовать данные термины.
4. Метод построения аспектной иерархии на основе пользовательских отзывов является популярным для исследований на английском языке. Имеется несколько альтернативных исследований и программ, работающих с англоязычным сегментом интернета, например, HASM (Hierarchical Aspect-Sentiment Model). Однако русскоязычный вариант метода не нашел отражения в обнаруженных источниках, поэтому в силу малого количества исследований в данной области есть основания утверждать, что подобное исследование в русскоязычном формате проводится впервые.

Данная работа была выполнена на основе статьи, чье название и авторы вынесены на слайд, в которой описан подход к организации иерархий аспектов для пользовательских отзывов об электронных устройствах в англоязычном сегменте. Стоит отметить, что статья была написана неподробно, в связи с чем многие важные для понимания логические переходы упущены. Именно поэтому подход в статье был взят за основу в данной работе и доработан, и подробно описан в дальнейшем с учетом всех недостатков.

1. В ходе работы были использованы технологии

Библиотека beautiful soup – парсинг

Библиотека Scikit learn - для работы с текстовыми и числовыми данными в ходе машинного обучения (например в рамках классификации One class svm или рандом форест)

1. Для построения иерархии требуется получить пользовательские отзывы. Самым удобным способом их получения является парсинг вебстраницы и сохранение отзывов в базу данных.

Изначально в качестве источника был выбран yandex market, однако предоставляемое сервисом API не позволяло получить требуемое для работы количество информации за заданный промежуток времени. Далее был опробован способ парсинга данных, но yandex market постоянно блокировал запросы от программы, в связи с чем было принято решение о поиске альтернативы.

Для работы был выбран сайт ulmart.ru, а именно раздел электронных товаров. Для получения отзывов со всех страниц товаров данного сайта был написан парсер, выявляющие отзывы и добавляющий их в базу данных. Парсер был выполнен с использованием библиотеки BeautifulSoup 4 версии [12]. Всего было получено 24093 отзывов.

1. На сайте юлмарт в разделе электронные товары имеется большое количество категорий, все они отражены на экране. Отзывы пользователей о товарах именно из этих категорий были получены методом парсинга.
2. Отзыв представляет собой объединение 3 частей: положительное резюме, отрицательное резюме и комментарий в свободном стиле, однако в работе данное разделение учтено не было, все остальные детали отзывов, а именно: вся информация о пользователе, рейтинг отзыва, количество согласных и несогласных людей, опыт использования пользователем продукта, были проигнорированы. Также была создана база данных с 415 идеальными аспектами – теми, что указываются, как описание характеристик того или иного товара.
3. О получении идеальных аспектов уже было сказано, это верхняя ветвь схемы на слайде. Для них была построена иерархия вручную. Теперь более подробно про нижнюю ветвь схемы. На ней представлены шаги, каждый из которых я более подробно объясню далее. Суть алгоритма такова, что из полученных отзывов было выявлено 45435 ключевых аспектов, для них рассчитаны семантические дистанции для каждой из пар и в итоге построена иерархия аспектов для отзывов пользователей об электронных устройствах на основе идеальной иерархии, построенной вручную.
4. Так как отзывы, полученные с ulmart.ru, приходят зашумленными и содержат в себе большое количество случайных символов, то первым делом производится очистка входных данных. В процессе очистки из отзывов исчезают случайные пользовательские повторы, лишние знаки препинания, символы, распознавание которых программой невозможно. Все отзывы были приведены к единому стилю, под этим подразумевается нижний регистр и обязательное наличие в конце точки, что способствовало корректной работе API ИСП РАН [2] в дальнейших вычислениях.

Далее производится выявление частей речи всех слов в каждом из отзывов. Данная обработка требуется для того, чтобы в дальнейшем осуществить поиск слова-существительного или набора слов, где главным словом также является существительное. Во втором случае также используется метод построения синтаксического дерева предложения, что позволяет найти нужные связи.

После создания набора всех аспектов производится выявление значимых аспектов в контексте данной работы. Для выполнения данной задачи были использованы возможности методов машинного обучения, а именно One-Class SVM. На вход модель получает набор из тренировочных данных, в результате полученный классификатор применяется к тестовым данным. Сначала все имеющиеся аспекты получают метку 1 или -1 в зависимости от того, присутствует ли данный аспект в списке идеальных аспектов. Далее данные делятся в соотношении 80 к 20 на тренировочные и тестовые. Набор из тренировочных данных очищается от тех аспектов, метка которых является -1. В итоге вся эта информация передается для обучения и предсказания классификатором меток для тестовых аспектов.

После данного шага было выявлено 421715 аспектов. Однако данное количество слишком велико для цели данной работы и содержит в себе большое количество мусора. Поэтому следующим шагом была группировка, под этим подразумевается удаление всех дублей, что существенно снизило количество аспектов и их стало 45435.

В дальнейшем для ускорения разработки программы количество аспектов было еще более сужено до 1000 экземпляров, которыми являются самые частые слова и словосочетания. Последующие результаты и числа будут актуальны в рамках данной 1000. Данный набор перекликается по содержанию с 415 идеальными аспектами, более 50% аспектов из них содержатся в суженом аспектном наборе.

1. Теперь перейдем к вычислению семантического расстояния. Для выполнения данной цели было выбрано 6 характеристик, которые вы видите на экране. Далее о каждой из них я расскажу подробнее.
2. PMI характеристика содержит в себе 2 характеристики. Разница в характеристиках заключается в корпусе отзывов/предложений, который передается для дальнейшей обработки в метод расчёта PMI. Формула для расчета на экране, ах и ау – это аспекты из пары.
3. Теперь пример, у нас имеется 2 аспекта в паре компьютер и экран и 3 изначальных отзыва пользователей до какой-либо обработки. Количество отзывов, где присутствуют оба аспектов всего 1. Количество отзывов, где есть аспект компьютер = 3, а экран = 1. Эти данные позволяют нам произвести вычисление характеристики pmi review. В случае с pmi sentence расчеты производятся аналогичным образом, только все количества мы берем относительно предложений, а не отзывов.
4. Contextual характеристика также содержит в себе 2 характеристики. Для локальной характеристики контекст - набор 2 левых и 2 правых слов от аспекта во всех отзывах, где он встречается. Для глобальной характеристики контекст - набор отзывов, где встречается конкретный аспект.
5. Теперь пример, все входные данные, как и в предыдущем примере. Количество слов в локальном контексте для аспекта всегда кратно 4, если слово стоит на границе, не имеет 2 слов слева или справа, то тогда подставляются слова \_BEGIN\_SENTENCE\_ и \_END\_SENTENCE\_. Для глобального контекста просто ищем отзывы, где встречается наше аспект.

Далее для каждого из контекстов, преобразованных в вектора с помощью vectorizer.fit\_transform, был произведен smoothing данных. Результатом работы метода smoothing’а данных является n-gram, каждое значение в котором, если он не пуст, делится на количество слов в контекстах аспектов. Далее для каждой пары аспектов производится вычисления значения kl-divergence между n-gram’ами с помощью встроенной функции scipy.stats.entropy. Данное значение и является результатом характеристики context. Разница лишь в том, где ищутся контексты для аспектов.

1. Далее syntactic feature. Для каждого из отзывов для ускорения работы программы с помощью API ИСП РАН для отзывов строятся синтаксические деревья. Для каждой пары аспектов выполняется поиск кратчайшего расстояния между ними в дереве.
2. Возьмем отзыв, который на слайде. Для примера приведен графический вариант синтаксического дерева для 1 предложения данного отзыва. Для общего дерева отзыва выполняется поиск кратчайшего пути между указанными аспектами. Если количество отзывов, где присутствует пара аспектов больше одного, то берется среднее арифметическое длин кратчайших путей, в противном случае просто берем данный путь. Это и будет значением величины syntactic.
3. Lexical характеристика самая простая в вычислении. Формула на слайде.
4. Пример на слайде
5. Все перечисленные мною характеристики важны и нужны для вычисления семантического расстояния между аспектами.
6. Для вычисления семантической дистанции было выбрано два способа расчётов, один из которых был приведен в статье, которая легла в основу данной работы, а второй является примером использования методов машинного обучения на практике. В первом способе производится вычисление вектора w, представляющего собой набор характеристик (PMI, Lexical, Syntactic, Contextual) идеальных аспектов. Данный вектор вычисляется по формуле со слайда. Далее для каждой пары аспектов семантическое расстояние равно формуле со слайда.
7. Для 1 подхода было решено также изучить влияние каждой из характеристик на итоговый результат для семантического расстояния. В итоге была получена таблица, которую вы видите на слайде. Как видно из приведенной таблицы, каждая из характеристик влияет на итоговый результат, что говорит о необходимости использования каждой из них. По очереди применялся алгоритм вычисления семантической дистанции между аспектами в парах для разных наборов характеристик. Также было попробовано добавление еще одной 7 характеристики, чье значение константно равнялось 1.
8. Второй способ представляет собой пример использования *RandomForestRegressor*, который являет частью пакета библиотеки *sklearn.ensemble*. В качестве входных данных для обучения модели поступают результаты идеальных аспектов, а именно их предвычисленные реальные длины путей для каждой пары аспектов, а также набор из вышеупомянутых 6 характеристик для каждой пары аспектов. Обучившись модель получает задание предсказать результаты уже для реальных данных.
9. Проведенное изучение двух подходов к вычислению семантического расстояния показало наличие, как преимуществ, так и недостатков у обоих. Дальше вся инфа на слайде.
10. В итоге мы получаем иерархию аспектов. На слайде приведен пример иерархии, на деле же иерархической дерево очень велико по размеру в связи с большим количеством аспектов. Поэтому приводится маленькая его часть.
11. На слайде список источник, использованных при работе.
12. Спасибо за внимание, я готова ответить на ваши вопросы.

RandomForestRegressor(criterion='mse', max\_depth=8, min\_samples\_leaf=10, n\_estimators=250)

RF (random forest)( Лео Брейманом и Адель Катлер) — это множество решающих деревьев. В задаче регрессии их ответы усредняются, в задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по следующей схеме:

* Выбирается подвыборка обучающей выборки размера samplesize (м.б. с возвращением) – по ней строится дерево (для каждого дерева — своя подвыборка).
* Для построения каждого расщепления в дереве просматриваем max\_features случайных признаков (для каждого нового расщепления — свои случайные признаки).
* Выбираем наилучшие признак и расщепление по нему (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), но в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщеплен

Одноклассный SVM используется для обнаружения новизны, то есть при заданном наборе выборок он обнаруживает мягкую границу этого набора, чтобы классифицировать новые точки как принадлежащие этому набору или нет. Класс, который реализует это, называется OneClassSVM. В этом случае, поскольку это тип неконтролируемого обучения, метод подгонки принимает только входной массив X, так как нет меток класса.