Санкт-Петербургский государственный университет
Кафедра математической лингвистики

Направление: «Лингвистика»
Образовательная программа: «Прикладная и экспериментальная лингвистика»
Профиль: «Компьютерная лингвистика и интеллектуальные технологии»

Автоматический анализ отзывов в рекомендательных системах

Выпускная квалификационная работа
студентки II-го курса магистратуры
Силифонтовой Натальи Александровны
Научный руководитель:
доцент, к. ф. н.
Хохлова Мария Владимировна

Санкт-Петербург
2017

Содержание

Введение

В последние несколько лет рекомендательные системы активно набирают популярность в сети Интернет. Впервые эта технология была опробована в области электронной коммерции, однако впоследствии стала применяться в различных сферах, среди которых – системы электронного образования (Глибовец, Сидоренко, 2012), новостные сайты, социальные сети. В первую очередь это связано с рядом преимуществ, включающих повышение эффективности поиска информации. В условиях увеличения роли Интернета в повседневной жизни, обучении, исследованиях, а также увеличения количества информации, доступной в сети, рекомендательные системы начинают играть всё большую роль. Чем больше данных имеется в нашем распоряжении, тем сложнее их обрабатывать и тем сложнее вычленить из этого потока информации те данные, которые действительно нужны. В связи с этим, в ситуации растущего информационного потока необходимость существования механизма, способного фильтровать данные из Интернета и увеличивать скорость обработки информации, становится неоспоримой.

Первые разработки рекомендательных систем относятся к началу 90-х годов (Adomavicius, Tuzhilin, 2005), однако тому, что их создание вышло на новый уровень, послужил конкурс Netflix Prize (Bennett, Lanning, 2007), организованный в 2006 году компанией Netflix (Глибовец, Сидоренко, 2012).

В своём «Руководстве по рекомендательным системам» Риччи, Рока и Шапира определяют рекомендательные системы (РС) как «инструменты и методы программного обеспечения, которые составляют предположения о том, какие объекты могут быть полезны пользователю. Эти предположения относятся к различным ситуациям, в которых нужно принять решение, например, какой товар купить, какую музыку послушать, какие онлайн-новости почитать.

«Объект» - это общий термин, который используется для обозначения того, что система рекомендует пользователям. Обычно РС фокусируется на определённом типе объекта (например, компактные диски или новости) и в соответствии с этим подбираются дизайн объекта, пользовательский

интерфейс и основной метод рекомендации, чтобы в результате обеспечить пользователя полезными и эффективными предположениями об этом типе объекта» (Ricci, Rokach and Shapira, 2015).

Рекомендательные системы позволяют решить проблему избытка информации путём предоставления персонализированных предположений, основанных на истории лайков и дислайков пользователя (Melville, Mooney and Nagarajan, 2002).

Одним из видов ресурсов, где наиболее часто используются рекомендательные системы, является онлайн-торговля. В своей статье «Рекомендательные приложения в электронной торговле» Шафер, Констан и Ридл говорят о том, что развитие электронной торговли и появление онлайн-магазинов позволило их владельцам предоставить пользователям более широкий выбор. Расширение выбора в свою очередь привело к увеличению количества информации, которую покупатель должен обработать, прежде чем выбрать то, что соответствует его нуждам. Чтобы справиться с этим избытком информации, в онлайн-магазинах применяют принципы массовой кастомизации[1] не к продуктам, а к тому, как они представлены в онлайн-магазине. Одним из способов достижения массовой кастомизации является использование рекомендательных систем (Shafer, Konstan, Riedl, 2001).

Таким образом, предполагается, что использование РС выгодно как для пользователей, так и для владельцев (и разработчиков) сайтов. Однако не всегда система может обеспечить пользователя хорошей рекомендацией, что может привести к тому, что пользователь перестанет верить системе или видеть в ней пользу и в конечном счёте компания может потерять клиента или получить негативные отзывы.

Целью данной работы является разработка механизма анализа отзывов, направленного на увеличение качества предоставляемых системой рекомендаций, а также на увеличение уровня доверия пользователей к системе.

---

[1] Массовая кастомизация – производство продуктов и услуг для узкой аудитории с учётом её интересов и требований (Петров, Петрова, 2016).

Для достижения этой цели требуется решить следующие **задачи**:

1. Проанализировать наиболее важные проблемы, возникающие при работе рекомендательных систем, и понять, как можно использовать инструмент анализа отзывов для их решения.

2. Сравнить основные типы рекомендательных систем, рассмотреть их преимущества и недостатки. Выбрать самые оптимальные алгоритмы.

3. Разработать методику анализа отзывов.

4. Разработать программу для автоматического анализа отзывов.

5. Проанализировать собранные отзывы с помощью разработанного инструмента.

6. Провести оценку полученных результатов.

Новизна данной работы заключается в следующем:

1. Совмещение метода, основанного на анализе нескольких атрибутов, и метода, основанного на анализе отзывов, при создании рекомендательной системы.

2. Анализ отзывов производится на основании упоминания определённого параметра в определённом блоке, а не на основании анализа тональности.

3. Система формирует рекомендации в виде подробного описания анализируемого продукта, выставляя ему оценки по каждому параметру. Система ориентирована на построение профиля продукта.

4. Словари, на основании которых производится анализ отзывов, были составлены с помощью инструмента “Sketch Engine” (https://the.sketchengine.co.uk).

Это исследование представляет **практическую значимость**, так как разработанная программа не имеет аналогов и при этом показывает достаточно хорошие результаты. Следовательно, использование разработанной технологии может способствовать увеличению качества работы рекомендательных систем, решению возникающих проблем и развитию сферы рекомендательных систем в целом. Результаты, полученные в ходе

проведённого исследования, можно использовать для составления рекомендаций других типов, отличных от приведённых в этой работе.

Глава 1. Основные проблемы рекомендательных систем

Чтобы уменьшить возможность появления ошибок при составлении рекомендаций, нужно иметь в виду, с какими проблемами чаще всего сталкиваются создатели РС и как предотвратить их возникновение.

В данном исследовании наибольший интерес представляет то, как можно решить возникающие проблемы с помощью автоматического анализа отзывов.

Одной из крупнейших проблем, возникающих при моделировании рекомендательной системы, является **недостаток данных**, так как для эффективной работы РС требуется огромный объём информации (Dehuri, 2012).

К сожалению, не всегда система может ненавязчиво получить от пользователя достаточное количество информации о его предпочтениях. В таком случае ей остаётся только обработать все возможные данные, которые уже имеются в её распоряжении. Основные типы данных, на которые опирается РС при составлении рекомендации: 1) Оценки (лайк или дислайк, по шкале от 1 до 5, и т.д.); 2) переходы по ссылке, время, проведённое на странице, купил пользователь этот товар или нет.

Тем не менее, в некоторых ситуациях данных критериев может быть недостаточно. К примеру, пользователь мог не поставить ни одного лайка или не купить ни одного товара за все свои посещения. В таком случае система проанализирует переходы по ссылке и время, проведённое на странице, но велика вероятность, что она может сделать неправильный вывод из этого анализа. Возможно, пользователь так много времени провёл на определённой странице, потому что писал длинный негативный отзыв о данном товаре. Однако, зачастую отзывы, которые оставляют пользователи, никак не интерпретируются системой (Schafer, Konstan, Riedl, 1999).

Ещё одной проблемой, с которой часто сталкиваются РС при составлении рекомендаций, является проблема **безопасности и мошенничества**. Рекомендация может быть составлена неверно в случае, если в её составление вмешивается человек, который искусственно завышает или занижает рейтинг того или иного объекта. Это может быть либо конкурент

фирмы-производителя оцениваемого объекта, либо наоборот её сторонник, в некоторых случаях это могут делать даже сами владельцы сайта, когда им требуется продать или прорекламировать тот или иной объект. Такие манипуляции могут вызвать недоверие к работе рекомендательной системы, а также уменьшить качество и точность рекомендаций (Khusro, Ali and Ullah, 2016).

В отношении этой проблемы использование содержания отзывов также может быть более полезным, чем использование оценок и переходов по ссылкам, по крайней мере, в случае, когда искусственное завышение рейтинга происходит при помощи компьютерной программы или робота. Отличить текст, написанный роботом, от текста, написанного человеком, гораздо проще, чем вычислить оценку или лайк, поставленный программой. Следовательно, хоть использование отзывов для составления рекомендаций и не поможет полностью решить эту проблему, по крайней мере, оно уменьшит её значимость, а значит, увеличит качество и точность рекомендаций.

Также при создании и использовании рекомендательных систем возникает проблема безопасности личных данных. Пользователи вряд ли захотят предоставить какие-либо данные системе, которая страдает от такой проблемы. Значит, РС должна вызывать доверие у пользователей, надёжно охраняя их личные данные (Khusro, Ali and Ullah, 2016).

Тем не менее, если мы говорим о таком источнике данных, как содержание отзывов, то здесь эта проблема уже не так актуальна. Для составления рекомендаций используется только та информация, которой пользователь намеренно решает поделиться. Однако, естественно, как и в случае с другими источниками, эти данные не должны быть использованы за пределами сайта.

Ещё одним из очевидных достоинств этого метода можно считать составление менее обобщённого мнения об объекте или предпочтения пользователя. Когда пользователь использует другие методы оценки объекта, он оценивает весь объект целиком, ставит ли он какое-либо количество звёзд

объекту или просто переходит по ссылке. Когда он оставляет отзыв об объекте, то чаще всего он не просто выражает своё мнение о нём в целом, а расписывает в подробностях его достоинства и недостатки. Когда объект можно описать с помощью набора атрибутов (цена, вес, цвет и т.д.), обычно используется метод основанной на предпочтениях оценки продукта. При таком подходе предпочтение пользователя может быть представлено в виде веса и/ или значения, которое придаётся каждому из атрибутов (Chen, Chen and Wang, 2015). Использование этого подхода приведёт к составлению более подробной, а значит, более точной рекомендации.

Как уже было упомянуто, при составлении рекомендации РС опирается на различные типы данных. Наиболее удобно использовать *оценки, получаемые от пользователей напрямую*, что включает в себя однозначные данные о том, к каким продуктам пользователи проявляют интерес. Например, в «Нетфликс» пользователи выражают своё мнение о фильмах с помощью звёздочек, а пользователи «ТиВо» выражают свои предпочтения в области ТВ-шоу, нажимая кнопки «палец вверх» или «палец вниз». Однако, получение прямой оценки не всегда возможно. Таким образом, РС могут также делать вывод о предпочтениях пользователя за счёт *косвенной* оценки, которая отражает мнение пользователя, основываясь на его поведении (Hu, Koren and Volinsky, 2008).

Во многих практических ситуациях большее значение для РС имеют именно косвенные данные. К примеру, когда пользователи неохотно выставляют оценки продуктам, или система ограничена в плане сбора прямой информации об их предпочтениях. При использовании косвенных данных, как только пользователь даёт согласие на сбор информации, более никакой дополнительной информации напрямую у него не запрашивается (Hu, Koren and Volinsky, 2008).

Очевидно, что использования только отзывов в качестве источника рекомендаций не всегда достаточно, по крайней мере, такой способ подходит не для всех типов рекомендаций и не для всех электронных ресурсов. Однако,

в большинстве случаев он может оказаться весьма полезным для рекомендательных систем в плане борьбы с вышеперечисленными проблемами, а также в плане увеличения точности и качества будущих рекомендаций.

Глава 2. Таксономия рекомендательных систем

В данной части работы описана таксономия рекомендательных систем, предложенная профессором Университета Миннесоты Джозефом А.

Констаном и профессором Университета штата Айдахо в Бойсе, Майклом Д. Экстрандом. В следующей части на основе предложенной таксономии разобран созданный механизм рекомендаций и его основные свойства.

Основные уровни классификации рекомендательных систем:

1. Сфера.
2. Цель.
3. Контекст рекомендации.
4. Чьи мнения ложатся в основу рекомендации.
5. Уровень персонализации.
6. Личная информация и степень доверия к системе.
7. Интерфейс.
8. Алгоритмы (Konstan and Ekstrand, 2013).

2.1. Сфера.

Электронная торговля. Наиболее популярной сферой деятельности, в которой активно используются РС, является сфера электронной торговли. Объектом рекомендации в таком случае являются продукты, изготовители или поставщики, наборы продуктов или пакеты услуг. Наиболее ярким примером такого ресурса является сайт Amazon.com.

Новости. Также рекомендательные системы часто используют на новостных сайтах. Объектом рекомендации для них являются новости и информация.

Музыка и видео. Рекомендательные системы также используются в музыкальных онлайн-плеерах, где в некоторых случаях система также учитывает порядок произведения композиций, и на веб-сайтах, предназначенных для просмотра видео.

Социальные сети и сайты знакомств. В последнее время рекомендательные системы всё чаще используются и в социальных сетях, таких как «Вконтакте», «Facebook» и т.д. Объектом рекомендации в социальных сетях могут выступать другие люди («Вы можете знать…»), группы и т.д.

2.2. Цель.

В большинстве случаев целью рекомендательной системы является либо *продажа,* либо *продвижение продукта*. Продуктом может служить не только

товар, но и, к примеру, видео или композиция какого-либо исполнителя. Существуют, однако, РС другого типа, которые способствуют образованию сообщества. Примером такой системы является OWL Tips, она помогает пользователям изучать более простые способы выполнения команд на компьютере.

Риччи, Рока и Шапира в своём руководстве перечисляют и другие причины, по которым люди используют данные технологии на своих сайтах. В конечном счёте, всё это помогает увеличить прибыль предприятия и сделать его лучше и удобнее для клиента.

1) Увеличение количества проданных продуктов.
2) Разнообразие продаваемых продуктов.
3) Увеличение уровня удовлетворённости клиентов.
4) Увеличение верности ресурсу.
5) Более точное понимание желаний пользователей (Ricci, Rokach and Shapira, 2015).

2.3. Контекст рекомендации.

Контекст определяется тем, при каких условиях у пользователя возникает потребность в рекомендации. Контекст учитывается на многих сайтах и во многих приложениях.

В литературе о контекстно-зависимых системах изначально контекст определялся как *местоположение пользователя, информация об окружающих людях и предметах, а также о происходящих с ними изменениях*. Впоследствии, были добавлены и другие факторы (Adomavicius and Tuzhilin, 2011).

К примеру, в приложении Foursquare РС учитывает местоположение пользователя, чтобы сначала включить в рекомендацию заведения, которые находятся к нему ближе всего. Если это сайт, где пользователь может забронировать отель, то РС зачастую учитывает цель его пребывания – работа или отдых. В случае онлайн-магазина в качестве контекста может учитываться то, покупает пользователь продукт для себя или в качестве подарка.

2.4. Чьи мнения ложатся в основу рекомендации.

Рекомендации для пользователей могут строиться на основе мнений *экспертов* в той области, в которой требуется рекомендация, или же на основе мнений *других пользователей*, вне зависимости от того, являются они экспертами в данной области или нет.

Например, на сайте “Wine.com” рекомендации о вине составляются экспертами. В системе “Phoaks-system”, напротив, рекомендации составляются с учётом мнений других пользователей. Система предоставляет пользователю список ссылок, напротив которых указано количество людей, которые ссылались на эту ссылку в данной теме. При желании пользователь может посмотреть текст сообщений, также он видит шкалу, на которой видно, как давно на этот ресурс ссылались (то есть, не устарела ли эта информация).

2.5. Уровень персонализации.

Самым простым типом является *неперсонализированная рекомендательная система.* Как и предполагает название, системы такого типа не учитывают личные предпочтения пользователей. Рекомендации, предлагаемые такой системой, идентичны для всех клиентов. Например, если мы откроем сайт Amazon.com как анонимный пользователь, мы увидим объекты, которые часто смотрят другие пользователи. Такие системы рекомендуют клиентам объекты, основываясь на том, что другие клиенты сказали об этих объектах или как их в среднем оценили (Poriya, Bhagat, Patel and Sharma, 2014).

Альтернативой такому типу рекомендательной системы является *персонализированная система.* При составлении рекомендаций такие системы могут опираться на различные факторы. К примеру, некоторые РС опираются на *демографические факторы* (возраст, пол и т.д.) и ориентируются на определённые целевые группы.

Другие системы персонализированного типа опираются на деятельность пользователя. Их можно разделить на *кратковременные*, то есть учитывающие текущую деятельность, и *долговременные*, то есть оценивающие долгосрочные интересы пользователя. К примеру, рекомендательная система на сайте “Brooks Brothers” базируется на личности пользователя – учитывает общую информацию о пользователе, основанную на его деятельности на других сайтах. А на ресурсе “CDNOW” пользователь может получить как кратковременные, так и долговременные рекомендации. В первом случае, система запрашивает у него имя одного исполнителя и, к примеру, один из его альбомов и на основании этих данных составляет для него рекомендацию. Однако, он может постоянно добавлять альбомы в список любимых альбомов и тогда система будет учитывать его долгосрочные интересы и их динамику для создания рекомендаций в будущем.

2.6. Личная информация и степень доверия к системе

Чтобы с наибольшей точностью предсказать, что хочет пользователь и в чём он нуждается, РС обычно опираются на огромное количество информации о пользователе, которая собирается иногда вопреки ожиданиям пользователя и тем самым затрагивает вопросы о *сохранности личной информации*. Возникновение таких вопросов, в свою очередь, может повлиять на мнение пользователей о системе. Такие данные о пользователе включают демографическую информацию, которая может указать на его личность (например, адрес электронной почты или номер социального страхования), а также данные, относящиеся к продуктам, которые пользователи оставляют в истории просмотров и истории покупок, что может указывать на их вкусы и привычки. В связи с различиями в уровне конфиденциальности фрагментов этой информации было бы не разумно применять единый механизм защиты данных, так как это может негативно сказаться на качестве рекомендаций. Следовательно, нужно разграничивать фрагменты информации о пользователе в соответствии с уровнем их конфиденциальности и учитывать это в контексте рекомендации; таким образом, разработчики смогут создавать такие рекомендательные системы, в которых сбалансированы уровень сохранности личных данных и качество рекомендации (Zhang, Wang and Jin, 2014).

Кроме того, с уровнем доверия к системе связан не только вопрос о безопасности личных данных. В поисках информации в Интернете мы часто сталкиваемся с *проблемой достоверности данных*. В связи с тем, что не существует какой-либо стандартной процедуры, которая помогала бы пользователям оценивать качество информации, им приходится работать в условиях неопределённости. Более того, существует вероятность столкновения с потенциальными проблемами, которые могут возникнуть вследствие следования рекомендациям: получение неверной информации может привести к потере времени или денег, или и того и другого. Следовательно, пользователи оказываются в зависимой позиции, когда обращаются к рекомендательной системе и к информации, которую она предоставляет, и им приходится обдумывать риск потери и возможность прибыли, прежде чем

принять решение поверить системе и перейти к покупке (Lenzini,Van Houten, Huijsen and Melenhorst, 2009).

Плюс ко всему, работа рекомендательной системы может вызывать сомнения у пользователя в связи с тем, что в большинстве случаев для получения стабильной прибыли от предприятия, а также для её увеличения, при составлении рекомендации нельзя обойтись без обращения к базовым законам бизнеса. Очевидно, что компания старается продвигать те товары, продвижение которых ей выгодно. Естественно, что это понимает и большинство пользователей, и так как рекомендательная система работает на владельцев ресурса и им во благо, зачастую пользователи могут относиться к её предположениям с осторожностью и сомнениями. Всё зависит от честности системы. Тем более, если владельцы предприятия будут слишком заметно вмешиваться в работу рекомендательной системы, то это внушит недоверие клиентам. Одним из стандартных примеров такого вмешательства (на взгляд автора, не превышающего границ разумного) может служить отсутствие в рекомендации товаров, которых нет в наличии. Такая ситуация не выгодна для предприятия, так как пользователь не сможет оформить заказ и возможно решит поискать такой же товар на другом сайте.

Подпортить репутацию системе могут и злоумышленники, то есть пользователи, которые искусственно завышают или занижают рейтинги тому или иному продукту. Бесплатный доступ, который предоставляют рекомендательные системы, подвергает их риску и возможности потенциальных нападений злоумышленников. Наиболее известные типы программ, которые используются для подобных спекуляций, – это “RandomBot” и “AverageBot”. При добавлении новых фейковых профилей, программе “RandomBot” необходимо немного информации о каждом пользователе, так как она создаёт случайные профили при помощи функции равномерного распределения. У программы “AverageBot” больше возможностей. Она создаёт фейковые профили на основе нормального распределения со средним значением по рейтингам всех пользователей.

Продвигая (или наоборот) целевой продукт, хакеры заставляют всех фейковых пользователей поставить данному продукту самую высокую (или самую низкую) оценку. Так как фейковые пользователи являются в системе близкими соседями настоящих пользователей (то есть, мнение первых учитывается при составлении рекомендации для вторых), то их оценки для целевых продуктов повлияют на рекомендации для настоящих пользователей (Zhang, Lee and Pitsilis, 2013).

2.7. Интерфейс

2.7.1. Тип выходных данных (вид рекомендации)

Существуют различные типы представления рекомендаций в рекомендательных системах. Всё зависит от того, как мы хотим, чтобы пользователь их использовал. Многие из основных типов рекомендаций, которые используются в современных рекомендательных системах, пришли из области традиционной (не электронной) торговли. В своей статье «Рекомендательные системы в электронной торговле» Шафер, Констан и Ридл выделяют семь типов рекомендаций:

1) *Просмотр данных.* В случае традиционной торговли клиент заходит в видео-магазин и просит продавца порекомендовать «комедию 50-х годов». В идеале, продавец порекомендует несколько фильмов и клиент пойдёт искать, где они расположены, просматривать обложки коробок и выбирать, что ему понравится. Однако, качество предоставленных рекомендаций зависит от знаний определённого продавца об огромном количестве фильмов. На сайте “Reel.com” есть некоторые преимущества, которые помогают сформировать карту фильмов для клиента. Во-первых, рекомендации нескольких продавцов/редакторов (в случае электронной торговли) комбинируются так, чтобы пользователю предоставлялись рекомендации как можно более высокого уровня вне зависимости от параметров запроса. Более того, рекомендации предоставляются сразу же со ссылками на рекомендованные продукты – нет необходимости искать рекомендованное по всему магазину. Такой тип рекомендаций помогает превратить пользователей, которые просто «зашли посмотреть» в покупателей. Это происходит за счёт того, что предоставляя хорошо организованные рекомендации, система помогает пользователям сузить их выбор и почувствовать больше уверенности относительно покупки какого-либо продукта (Schafer, Konstan and Riedl, 1999).

2) *Похожий продукт.* Тип рекомендации, который, как и предыдущий, пришёл из традиционной торговли. Одним из явных примеров такой

рекомендации является блок «С этим также покупают» на многих сайтах электронной торговли, к примеру, на “Amazon.com”.

Отображённые продукты могут быть полностью отобраны на основании продукта или продуктов, к которым пользователь показывал интерес. Применяя рекомендации такого типа, сайты знакомят пользователя со своей линией продуктов и в идеале могут продать большее количество продуктов за один заказ (Schafer, Konstan and Riedl, 1999).

3) *Электронное письмо.* В том случае, если пользователь просматривал на сайте объекты определённого типа, но ничего не купил, система отправит ему рекомендацию на электронную почту. Также существуют ситуации, когда продукта, который пользователь хотел приобрести, нет в наличии. В таком случае, система отправит письмо пользователю, как только данный товар появится в наличии (Schafer, Konstan and Riedl, 1999).

4) *Текстовые комментарии.* Постепенно всё больше веб-сайтов используют текстовые комментарии, которые оставляют клиенты, для того, чтобы обеспечивать пользователей рекомендациями. Функция пользовательских комментариев на сайтах “Amazon.com”, “Flipkart.com”, “Snapdeal.com” и “eBay” помогает формировать «мнение со стороны», а также позволяет клиентам найти интересующий продукт и просмотреть комментарии об этом продукте от других клиентов. Это помогает сайтам зарабатывать деньги, так как вселяет в пользователей уверенность в продаваемых товарах или услугах – идея в том, что если достаточно много людей утверждает, что продукт хорош, а продавцу можно доверять, то, скорее всего, так и есть. Это не только помогает превратить посетителей в покупателей, но также увеличивает их верность сайту. Когда пользователи понимают, что можно доверять этим непредвзятым рекомендациям, то они с большей вероятностью вернутся к ним, когда им потребуется помощь в принятии какого-либо решения (Hooda, Singh and Dhawan, 2014).

5) *Средний рейтинг.* Более простой доступ к такому «мнению со стороны» обеспечивает средний рейтинг продукта. Вместо того чтобы

просматривать большое количество текстовых комментариев, можно просто собрать оценки пользователей в численном виде. Собирая оценки пользователей и высчитывая на их основе среднюю оценку продукта, система предоставляет пользователям возможность получить общие сведения о качестве продукта буквально за одну секунду. Так же, как и в случае с текстовыми комментариями, средний рейтинг продукта помогает превратить посетителей в покупателей и увеличить их верность ресурсу (Schafer, Konstan and Riedl, 1999).

6) *Топ-N лучших продуктов*. Система подбора совпадений для книг сайта “Amazon.com”, подборщик стиля сайта “Levi’s” и функция “My CDNOW” на сайте “CDNOW” в числе прочих используют рекомендации с помощью списка топ-N лучших продуктов. Как только система получает какие-либо данные о том, что нравится или не нравится клиенту, она может предоставить ему персональный список нескольких наиболее подходящих для него продуктов (Schafer, Konstan and Riedl, 1999).

7) *Упорядоченные результаты поиска.*

И последний тип вывода данных – упорядоченные результаты поиска. По сути, это одна из разновидностей уже упомянутого списка нескольких лучших продуктов. Только этот вариант менее ограничен.

В то время как список N лучших товаров ограничивает количество рекомендаций, данный тип вывода позволяет пользователю продолжить поиск среди прочих вариантов, которые вероятно могут его заинтересовать. К примеру, функция “We predict” на сайте “Moviefinder.com” предоставляет пользователям результаты поиска, упорядоченные в зависимости от вероятности того, что пользователю понравится этот продукт. Как и в случае с предыдущими типами вывода данных, этот помогает превратить посетителей в покупателей (Schafer, Konstan and Riedl, 1999).

2.7.2. Тип входных данных (на основании чего строится рекомендация)

Если говорить обобщённо, то по типу входных данных, то есть по типу информации, которая кладётся в основу будущей рекомендации, рекомендации делятся на явные и неявные.

Явные рекомендации основываются на оценке продуктов, полученной напрямую от пользователя. Данные такого типа система получает, когда напрямую запрашивает у пользователя мнение об объектах. Неявные рекомендации создаются на основе данных о поведении пользователя, то есть, к примеру, о покупках, просмотрах, или на данных о том, какие книги он берёт в библиотеке (Neumann, 2009).

Однако, на основании типа входных данных рекомендательные системы можно классифицировать и более подробно.

В зависимости от источника входных данных рекомендательные системы можно разделить на три категории. РС первого типа в качестве входных данных используют только информацию об истории поведения пользователя, для которого создаётся рекомендация, и информацию об объекте, который является кандидатом для рекомендации, не придавая значения истории действий других пользователей.

РС второго типа учитывает информацию о действиях всех пользователей в данном приложении, не ограничиваясь информацией о действиях только одного пользователя. Этот тип рекомендательных систем изучает информацию более глубоко и более широко в сравнении с системами первого типа.

Системы третьего типа были представлены не так давно и их появление связано с развитием различных социальных сетей (Facebook, LinkedIn, RenRen, Weibo, и т.д.). Целью РС этого типа является использовать связи между пользователями в этих социальных сетях для создания рекомендаций. Принципом для РС такого типа является: «Рекомендация от друга – лучше другой рекомендации» (Lei, Qin and Zhang, 2014).

2.8. Алгоритм

Как уже было сказано ранее, рекомендательные системы могут быть персонализированными и не персонализированными. В зависимости от принадлежности РС к тому или иному типу, при построении рекомендации они используют различные алгоритмы.

2.8.1. Алгоритмы рекомендаций в не персонализированных рекомендательных системах

2.8.1.1. Метод обобщённого мнения (Aggregated Opinion Approach)

Существует большое количество сайтов, которые используют не персонализированные РС, отображая среднее от всех пользовательских оценок. Зачастую такой тип рекомендации используют онлайн гиды по ресторанам. Система показывает ресторан и его оценку, которая равна среднему значению от всех оценок, которые поставили данному ресторану другие пользователи. Некоторые онлайн гиды предоставляют пользователям возможность оценить ресторан по четырём критериям (Еда, Обстановка, Цена, Количество). Обычно в диапазоне от 0 до 5. Затем высчитывается средняя оценка и впоследствии отображается рядом с названием ресторана (Poriya, Bhagat, Patel and Sharma, 2014).

2.8.1.2. Метод ассоциации продуктов (Product Association Approach).

Этот алгоритм учитывает процент пользователей, которые поставили оценку данному продукту. Этот метод – самый эффективный для не персонализированных РС. Большинство онлайн магазинов используют этот метод с помощью блока «Люди, которые купили товар 1, также купили товар 2». При составлении такого блока используется техника анализа потребительской корзины. Рекомендации в данном случае составляются на основании того, что пользователь делает на данный момент, то есть, какие продукты он просматривает или покупает, и на основании того, что сейчас у него в корзине (Pereira and Varma, 2016).

Для составления рекомендаций такого типа требуется сразу же исключить из входных данных алгоритма множества, которые встречаются

очень редко (к примеру, сочетание {моторное масло, тушь для ресниц}). Существуют различные алгоритмы для анализа предпочтений покупателей и выявления наиболее часто встречающихся наборов. Наиболее распространённым алгоритмом для составления правил ассоциации продуктов является априорный алгоритм (лат. a priori — буквально «от предшествующего» — знание, полученное до опыта и независимо от него), который в 1994 г. предложили R. Agrawal и R. Srikant[2]. В основу алгоритма заложены данные о частоте выбора покупателями определённых сочетаний продуктов.

Этот алгоритм имеет два параметра, support (англ.: support – опора, поддержка) и confidence (англ.: confidence – уверенность, доверие). От значения этих параметров зависят искомые правила ассоциации. Они отражают полезность и точность правил соответственно. Правила ассоциации выявляются на основании базы данных, каждая запись в которой представляет собой набор продуктов. Для ассоциативных правил вида $A \Rightarrow B$, где A и B – это сочетания продуктов, формулы поддержки (Supp) и уверенности (Conf) определяются следующим образом:

$$\mathrm{Supp}(A \Rightarrow B) = \frac{\text{записи, содержащие и A и B}}{\text{общее количество записей}}$$

$$\mathrm{Conf}(A \Rightarrow B) = \frac{\text{записи, содержащие и A и B}}{\text{записи, содержащие A}}$$

Правила ассоциации, которые удовлетворяют как минимальному установленному значению уверенности, так и минимальному установленному значению поддержки, называются сильными правилами и считаются заслуживающими внимания (Bendakir and Aïmeur, 2006).

[2] R. Agrawal and R.Srikant. “Fast algorithms for mining association rule” in Proceedings of the 20th International Conference on Very Large Databases, pp. 487-499, 1994.

2.8.2. Алгоритмы рекомендаций в персонализированных рекомендательных системах

В различных статьях и научных работах авторы не всегда одинаково классифицируют рекомендательные системы на основании алгоритма рекомендации. Однако по большей части эти различия незначительны. К примеру, по мнению доктора физико-математических наук из Московского государственного университета им. М.В. Ломоносова, А.Г. Дьяконова, методы, применяющиеся при разработке РС, можно разбить на две основные группы: методы коллаборативной фильтрации (collaborative filtering) и контентные методы (content-based, information filtering). В случае методов коллаборативной фильтрации, при разработке используются статистические данные о поведении пользователей (например, РС рекомендует продукты, которые были интересны для похожих пользователей). При разработке РС, ориентирующихся на контентные методы, используются какие-либо характеристики продуктов (например, рекомендуются товары из той же ценовой группы, той же категории и т.д.). Естественно, возможно и применение гибридного подхода, то есть использование в работе одной РС и тех и других методов (Дьяконов, 2012).

Рассмотрим вышеперечисленные методы более подробно.

2.8.2.1. Контентный метод или фильтр информации, основанный на содержании (content-based information filtering)

При использовании данного метода в системе формируются профили пользователей и объектов на основании анализа текстовой метаинформации объектов. После этого, с помощью какой-либо меры близости, часто – коэффициента Пирсона, выделяются объекты, по своим характеристикам наиболее близкие к профилю пользователя (Федоровский, Логачева, 2011).

Например, у фильмов есть какие-то определённые важные характеристики, такие как жанр, актёры, режиссёр, и когда пользователь выставляет оценку какому-либо фильму, тем самым он обновляет модель предпочтений. Эта модель описывает то, какие характеристики фильмов

предпочитает данный пользователь, и часто представляет собой таблицу, на которую система ссылается как на вектор ключевых слов или тестовый вектор. Каждый раз, когда пользователь оценивает какой-либо фильм, информация в модели предпочтений обновляется (Kulkarni, Wagh, Badgujar and Patil, 2016).

Достоинства данного метода: РС может составлять рекомендации для незнакомых пользователей (т.е. пользователей, которые ничего ещё не оценивали), таким образом, вовлекая их в сервис. Недостатки: понижается уровень точности предоставляемых рекомендаций, немного увеличивается время разработки (Савкова, Привалов и Чепикова, 2016).

2.8.2.2. Метод, основанный на знаниях (knowledge-based)

При использовании данного метода, РС составляют рекомендации, основанные на информации о предметной области. Зачастую предыдущий метод считают подвидом метода, основанного на знаниях, так как в контентном методе в качестве знаний можно рассматривать информацию о продукте (Савкова, Привалов и Чепикова, 2016).

Однако, не все авторы придерживаются такого мнения, так как у метода, основанного на знаниях, есть свои отличительные черты. Использование данного метода очень хорошо подходит для рекомендаций тех продуктов, которые не покупают (или не просматривают) на регулярной основе.

В связи со сложностью выбора или с серьёзностью подхода пользователя к приобретению продукта в тех сферах, где используется основанный на знаниях подход, пользователи, как правило, проявляют большую активность в выражении своих требований по отношению к искомому объекту. К примеру, в случае с выбором фильма пользователь примет рекомендацию зачастую с большим удовольствием и с меньшим количеством уточнений о качествах искомого объекта, чем в случае с выбором машины или квартиры.

Иначе говоря, значительным отличием систем такого типа является наибольший уровень контроля пользователя над процессом составления рекомендаций. Плюс ко всему, РС, основанные на содержании или использующие метод коллаборативной фильтрации, принимают во внимание в большей степени информацию о предыдущей деятельности пользователя, тогда как РС, основанные на знаниях, ориентируются на то, что пользователь хочет в данный момент (Aggarwal, 2016).

2.8.2.3. Метод коллаборативной фильтрации (collaborative filtering).

По мнению некоторых исследователей в области рекомендательных систем, наиболее эффективным методом персонализированного информационного фильтрования является коллаборативная фильтрация.

При использовании этой техники, РС рекомендует пользователю объекты на основании данных о взаимодействии с объектом других пользователей. При построении рекомендаций с помощью этого метода используются известные оценки (мнения) группы пользователей для прогнозирования неизвестных предпочтений другого пользователя. Основной принцип коллаборативной фильтрации состоит в предположении о том, что вероятно те, кто похожим образом оценивали какие-либо объекты в прошлом, будут впоследствии так же оценивать другие объекты (Ролгин, 2016).

На сегодняшний день разработано множество алгоритмов коллаборативной фильтрации, которые можно разделить на две основные группы:

1) Методы, основанные на анализе имеющихся оценок, – анамнестические методы (Memory-based).

Работа алгоритмов этой группы основывается на статистических методах, которые позволяют выявить группу пользователей наиболее близкий к целевому пользователю. Другое название этого метода – *метод ближайших соседей (Neighbourhood-based)*. При работе РС используются предшествующие оценки, сделанные клиентов, и анализируются оценки других клиентов с подобными предпочтениями. Рекомендации для целевого пользователя составляются на основании вычисления меры похожести по всем собранным данным (Пятикоп, 2013(а)).

Рекомендации, получаемые в результате использования данного метода, можно разделить на два вида: *основанные на сходстве пользователей (User-based)* и *основанные на сходстве объектов (Item-based)*.

Системы, ориентирующиеся на сходство пользователей, такие как “GroupLens”, “Bellcore video” и “Ringo” делают выводы об интересе

пользователя u к объекту i, используя оценки, которые поставили этому объекту другие пользователи, которые называются соседями, и имеют похожие паттерны оценивания. Соседями пользователя u, как правило, являются пользователи v, оценки которых для объектов, оцененных как пользователем u, так и пользователями v, то есть I_{uv}, наиболее схожи с оценками пользователя u (Desrosiers and Karypis, 2011).

Алгоритм составления рекомендаций, *основанных на сходстве пользователей*, включает в себя 3 этапа:

1. Для каждого пользователя u вычисляется, насколько его предпочтения совпадают с предпочтениями пользователя a.
2. Затем, выбирается множество пользователей, наиболее близких к пользователю a.
3. Формируется оценка для определённого объекта i на основе оценок этого объекта «соседями», выбранными на прошлом этапе (Гомзин, Коршунов, 2012).

При построении рекомендаций второго типа, то есть основанных на сходстве объектов, предполагается, что если два объекта получили одинаковые оценки от пользователей, то они похожи, и, следовательно, у пользователей будут аналогичные предпочтения по отношению к подобным объектам.

Преимуществом данного подхода по сравнению с предыдущим является возможность вычисления степени близости рассматриваемого объекта ко всем остальным в отложенном режиме. Это означает, что процесс формирования рекомендаций может быть разделен на два этапа: отложенная стадия – вычисление близости объектов по отношению друг к другу, и стадия в реальном времени – вычисление рейтингов рассматриваемых объектов. В связи с этим, данный алгоритм более эффективен с точки зрения времени составления рекомендаций (Пятикоп, 2013(b)).

Алгоритм работы рекомендательной системы, *основанной на сходстве объектов*, почти такой же, как и в случае со сходством пользователей:

1. Для каждого объекта система вычисляет, насколько он похож на объект i, для которого предсказывается пользовательская оценка.
2. Происходит выбор объектов, наиболее похожих на i.
3. Система предсказывает оценку данному объекту на основе оценок, полученным в результате второго этапа от пользователя a (Гомзин, Коршунов, 2012).

Для выполнения первого пункта из описанных выше алгоритмов, то есть для вычисления степени близости, Гомзин и Коршунов предлагают использовать либо косинус между векторами (вектор – строка пользовательских оценок, либо строка оценок объекту, в зависимости от выбранного метода), либо коэффициент корреляции Пирсона (Гомзин, Коршунов, 2012). Оба эти способа можно использовать как для вычисления близости между пользователями, так и для вычисления близости между объектами.

Однако существуют и другие способы, которые используются в РС для вычисления степени близости пользователей или объектов:

- расстояние Эвклида, Хемминга;
- ранговая корреляция Спирмена;
- коэффициент Жаккара (Пятикоп, 2013).

2) Методы, основанные на анализе модели данных, – модельные методы (Model-based).

Разработка моделей (машинного обучения, алгоритмы глубинного анализа данных) позволяет системе научиться распознавать сложные паттерны на основании тренировочных данных, и впоследствии формировать толковые рекомендации для задач коллаборативной фильтрации, работая с тестовыми данными или уже с реальными данными, основываясь на выученных моделях. К алгоритмам, основанным на анализе модели данных, относятся методы Байесовских сетей, методы кластерного анализа, методы, основанные на Марковских моделях (MDP), методы латентного семантического анализа (LSA), метод сингулярного разложения (SVD) и анализ главных компонент

(PCA). Эти алгоритмы были предложены для решения проблем, возникающих при использовании анамнестических методов (Su and Khoshgoftaar, 2009).

Во многих случаях идея модельных методов, а также одно из основных преимуществ методов этого типа, заключается в том, что с их помощью можно построить модель, которая будет меньше по объёму занимаемой памяти. Плюс ко всему, использование моделей позволяет учитывать не только похожесть объектов, но и их ценность (Гомзин, Коршунов, 2012).

Однако, несмотря на некоторые преимущества, у методов этого типа есть и существенные недостатки. Во-первых, модель требует значительно большего времени для вычисления, во-вторых, требует повторного вычисления, как только обновляется матрица данных, что происходит каждый раз, когда пользователь снова оценивает какой-либо объект. В основном, незначительные изменения в матрице данных не учитываются, тем не менее, как только их становится больше, модель нужно тренировать по-новому (Levinas, 2014).

2.8.2.4. Рекомендательные системы, основанные на нескольких критериях, или мультиатрибутивные рекомендательные системы (multi-criteria recommender systems).

Этот тип рекомендательных систем можно назвать более инновационным по сравнению с предыдущими классическими подходами. При применении традиционных подходов, таких как коллаборативная фильтрация и подход, основанный на содержании, учитываются только общие оценки объектам, что, однако, не может пояснить, почему пользователи поставили такие оценки. К примеру, два пользователя одинаково оценили один и тот же фильм, но по разным причинам: одному нравится его жанр и режиссёр, а другому нравится актёр и актриса. Таким образом, всё большее количество работ, написанных в последние годы, были направлены на то, чтобы распределить пользовательские оценки по атрибутам объектов и в результате были разработаны так называемые мультиатрибутивные РС. К примеру, в работе Адомавициуса и Квона (Adomavicius and Kwon, 2007) классический алгоритм коллаборативной фильтрации был усовершенствован с помощью использования установленных пользователем мультиатрибутивных оценок для вычисления близости между пользователями. Они также предложили вычислять общий рейтинг объекта с помощью функции объединения оценок, данных отдельным его атрибутам. Этот подход показал себя лучше по сравнению с традиционным методом коллаборативной фильтрации, основанном на общей оценке (Wang and Chen, 2012).

2.8.2.5. Рекомендации, основанные на пользовательских отзывах.

Ещё одним из более современных методов составления рекомендаций в РС является метод, основанный на использовании отзывов. При использовании этого метода система учитывает пользовательские отзывы о продуктах, однако изначально в своей основе он лишь подразумевал улучшение традиционного метода коллаборативной фильтрации с помощью формирования одномерных виртуальных рейтингов на основании результатов классификации тональности отзывов (Poirier, Tellier, Fessant, and Schluth, 2010; Zhang, Ding, Chen, Li and Zhang, 2012). Только в некоторых работах был тщательно проработан эффект использования многомерной оценки тональности на уровне атрибутов объектов. К примеру, в своей работе об использовании текстовых отзывов для составления рекомендаций Якоб, Вебер, Мюллер и Гуревич предложили использование метода многомерного разложения матрицы (Jakob, Weber, Müller, and Gurevych, 2009), для того, чтобы смоделировать отношения между пользователями, фильмами и мнениями с точки зрения отдельных характеристик (Wang and Chen, 2012).

Техники коллаборативной фильтрации показывают очень хорошие результаты, когда система располагает достаточным количеством информации о рейтингах. Однако в условиях проблемы разреженности рейтингов, с которой часто сталкиваются РС, их эффективность падает в связи с низким покрытием пространства для рекомендаций или со сложностями в плане предоставления пользователям возможности выражения своих предпочтений в виде линейных рейтингов. Для того чтобы справиться с этой проблемой был разработан метод рекомендаций, основанных на содержании, при применении которого система опирается на содержание объектов, и на его основании находит объекты, содержание которых похоже на содержание тех, которые понравились данному пользователю. В некоторых исследованиях было предложено использование других типов получаемой от пользователя информации, например, использование тэгов (произвольно выбранные/ написанные ключевые слова) (Marinho et al., 2011; Zhao, Du, Nauerz, Zhang,

Yuan and Fu, 2008), а также использование социальных отношений (таких как дружба, принадлежность к какой-либо организации, доверительные отношения) для увеличения точности рекомендаций. Однако этих методов всё также недостаточно, особенно в тех случаях, когда о деятельности какого-либо пользователя мало данных. Их эффективность также ограничена в условиях высокого уровня общей разреженности данных (Chen, Chen and Wang, 2015).

В (Chen, Chen and Wang, 2015) авторы разделяют исследования в сфере РС, основанных на отзывах, на две основные категории в зависимости от того, с какой целью используются отзывы:

А) Основанное на отзывах моделирование пользовательского профиля.

Б) Основанное на отзывах моделирование профиля продукта (Chen, Chen and Wang, 2015).

Исследования из первой категории они, в свою очередь, разделяют на подгруппы в зависимости от того, на каком типе пользовательского профиля авторы исследования акцентируют своё внимание: профиль, основанный на ключевых словах, то есть профиль, который строится с помощью извлечения из отзывов часто употребляемых терминов; основанные на профиле рейтингов, то есть те, где отзывы используются либо для того, чтобы сделать вывод о том, как бы пользователь оценил какой-либо объект (когда он не выставлял оценки), либо для дополнения существующих оценок; основанные на предпочтениях в отношении каких-либо характеристик, в которых в отличие от исследований, основанных на рейтингах, анализируется не то, насколько пользователю понравился какой-либо объект, а то, почему он ему понравился (Chen, Chen and Wang, 2015).

Исследования из второй категории авторы разделили на две подгруппы, в зависимости от того, на каком типе мнения акцентируется внимание при составлении рекомендации: основанные на мнениях об атрибутах продукта; основанные на сравнении пользователем одного продукта с другим по его атрибутам (Chen, Chen and Wang, 2015).

2.8.3.Гибридные рекомендательные системы.

Гибридные РС совмещают в себе два или более метода рекомендаций для достижения лучших результатов и решения проблем, возникающих при применении только одного метода. Чаще всего используется метод коллаборативной фильтрации и какая-либо другая техника для того, чтобы избежать возникновения проблем при увеличении нагрузки на систему (Burke, 2002).

Во многих работах о гибридных системах авторы акцентируют своё внимание на совместном применении метода коллаборативной фильтрации и метода, основанного на анализе содержания (content-based filtering) (Shih and Liu, 1999; Нефёдова, 2012). Однако в рамках данной работы больший интерес вызывают гибридные РС, которые используют анализ отзывов.

В (Soni, Goyal, Vadera and More, 2017) авторы рассматривают исследования в области рекомендательных систем, которые рекомендуют фильмы, с точки зрения использования в этих системах анализа пользовательских отзывов.

По их мнению, инструмент рекомендации, совмещающий использование таких алгоритмов, как метод коллаборативной фильтрации, контентный метод, и метод, основанный на контексте (Pathak, Matharia and Murthy, 2013), может предоставить, в большинстве случаев, точные рекомендации . Тем не менее, его применение ограничено тем, что в некоторой степени, если фильм был хорошо оценён публикой, он это игнорирует, а также не принимает во внимание текстовые комментарии, которые во многих случаях формируют более точное представление о фильме.

Техники глубинного анализа текста (Liu, Hsaio, Lee, Lu and Jou, 2012), в свою очередь, анализируют только оценки и отзывы для того, чтобы сделать вывод о том, был фильм принят хорошо или плохо, вводя такой параметр, как оценка полярности (Polarity Score). Эта техника хорошо подходит для того, чтобы получить общее мнение о фильме, но она не была должным образом интегрирована в рекомендательную систему.

Рекомендательная система, которая учитывает такие элементы как, информация о фильме, общий рейтинг фильма и собранные отзывы, стала бы гораздо более подходящим ресурсом для создания рекомендательной системы в области кинематографии (Soni, Goyal, Vadera and More, 2017).

Таким образом, авторы упомянутой работы считают, что создание гибридной рекомендательной системы, одним из компонентов которой является инструмент анализа отзыва, может заметно улучшить уровень получаемых рекомендаций.

Глава 3. Анализ созданного механизма рекомендаций на основе вышеизложенной таксономии

Разработанный механизм рекомендаций не является в полной мере рекомендательной системой, скорее он может быть использован как часть гибридной РС, которая направлена на увеличение качества рекомендаций и дополняет другие её компоненты. Однако, несмотря на то, что механизм не является рекомендательной системой, его можно рассмотреть в рамках изложенной ранее классификации.

1. Сфера.

Механизм анализа отзывов был разработан для секции «Ноутбуки» с сайта «Яндекс Маркет», поэтому он относится к сфере электронной торговли.

2. Цель.

В связи со сферой, для которой был разработан механизм, можно считать, что его целью является продажа продукта. Однако более точным определением является скорее помощь в принятии решения, и только как следствие – продажа.

Кроме того, целью создания данного механизма также можно считать увеличение уровня доверия пользователей к системе, и в связи с этим – увеличение верности ресурсу и улучшение его репутации. Доверие пользователей к получаемым рекомендациям может возрасти в связи с тем, что рекомендации строятся на основании отзывов других пользователей, которые уже купили продукт, и, для которых зачастую (или по мнению пользователей) нет никакой выгоды в том, чтобы искусственно завышать или занижать рейтинг продукта.

3. Контекст.

Контекст рекомендации данный инструмент не учитывает.

4. Чьи мнения ложатся в основу рекомендаций.

При построении рекомендаций с помощью разработанного механизма учитываются мнения других пользователей, чаще всего это пользователи, которые приобрели данный продукт.

5. Уровень персонализации.

Механизм предоставляет рекомендации одинаковые для всех пользователей, поэтому, скорее всего, правильнее будет отнести его к неперсонализированным РС. Однако при составлении этих рекомендаций он учитывает мнение каждого пользователя, более того, мнение каждого пользователя о каждой характеристике продукта, в связи с этим его можно также отнести и к персонализированным РС.

В принципе, РС, основанные на отзывах, другие авторы зачастую относят к персонализированным (Jain, Jain and Kapoor, 2016; He, Chen, Kan, and Chen, 2015), так как они, как правило, используются для составления персонализированных рекомендаций, но в данном случае это не совсем верно. Плюс ко всему, созданный механизм не подходит полностью ни под одно из описаний, изложенных в таксономии. В связи с этим, мы считаем целесообразным отнести его к гибридным РС.

В дальнейшем результаты, полученные в ходе работы механизма анализа отзывов, планируется использовать также и для составления персонализированных рекомендаций.

6. Личная информация и степень доверия к системе.

Личная информация пользователей при работе данного механизма не разглашается. Для составления рекомендаций используются только те данные и мнения, которыми пользователь счёл разумным поделиться с другими пользователями, то есть мнения, описанные в его отзывах.

Как уже было сказано ранее, мы предполагаем, что использование такого типа рекомендаций приведёт к увеличению уровня доверия к системе, так как пользователи точно будут знать, на основании чего формируются эти рекомендации, и конечно же, они всегда смогут сверить рейтинги, полученные в результате работы программы с отзывами к данному продукту, чтобы убедиться в честности системы.

7. Интерфейс.

1. Тип выходных данных (вид рекомендации).

По типу выходных данных разработанный инструмент можно отнести к РС, предоставляющим в качестве рекомендации средний рейтинг продукта. Однако в связи с тем, что данный механизм относится к мультиатрибутивным РС, то он предоставляет средний рейтинг не для всего продукта, а средний рейтинг для каждого из его атрибутов.

2. Тип входных данных (на основании чего строится рекомендация).

По типу входных данных инструмент можно отнести к РС, использующим явные данные, то есть данные полученные от пользователя напрямую.

Для составления рекомендаций программа использует данные, полученные от всех пользователей, когда-либо оценивших анализируемый продукт с помощью отзыва.

2. Алгоритм.

Система относится к РС, основанным на пользовательских отзывах. При этом, она также является мультиатрибутивной, так как при проводимом анализе делается вывод не о мнении пользователя о продукте в целом, а о его мнении о каждом из атрибутов продукта.

На основании классификации, предложенной в (Chen, Chen and Wang, 2015), созданный механизм можно охарактеризовать как а) ориентированный на создание профиля продукта и б) основанный на мнениях об атрибутах продукта.

В ходе данного исследования был создан только один компонент РС, то есть инструмент автоматического анализа отзывов. В будущем планируется интегрировать разработанный инструмент в гибридную рекомендательную систему, которая будет использовать и другие методы рекомендаций в своей работе.

Глава 4. Исследование

Разработка программы, которая будет анализировать отзывы пользователей и на основании результатов проведённого анализа составлять рекомендации в виде оценок для каждого параметра исследуемого товара.

Этапы исследования:

1. Сбор отзывов с сайта.
2. Составление словарей, на основании которых будет проводиться анализ отзывов.
3. Разработка программы.
4. Оценка результатов.

Рассмотрим перечисленные этапы более подробно.

4.1. Сбор отзывов с сайта

В качестве материала для эксперимента были использованы отзывы с сайта «Яндекс Маркет» из секции «Ноутбуки». Этот сайт был выбран, так как отзывы на нём поделены на три блока: «Достоинства», «Недостатки» и «Комментарий». Этот факт упрощает работу программы, так как ей нет необходимости анализировать то, как именно оценил пользователь тот или иной параметр продукта, достаточно просто того, в каком блоке он его упомянул. Естественно, существует возможность того, что пользователь может описать какое-либо достоинство продукта в блоке «Недостатки», или наоборот, но такая вероятность достаточно мала, и ей можно пренебречь. Блок «Комментарий» программа не учитывает, так как в большинстве случаев в этом блоке пользователи просто повторяют то, что уже описано в других блоках, только более подробно. Для составления рекомендаций эти детали не существенны.

Секция «Ноутбуки» была выбрана в связи с тем, что этот тип товара достаточно популярен, при этом относится к той категории товаров, в отношении которых пользователям зачастую требуется помощь в принятии решения. Следственно, при выборе ноутбука они часто обращаются к рекомендациям.

На основании этих отзывов было составлено две выборки – тренировочная и тестовая.

Тренировочная выборка легла в основу двух корпусов – «Достоинства» и «Недостатки». Эти корпусы были использованы для составления словарей, на которые опирается программа при анализе отзывов. Объём выборки составил 600 отзывов. Отзывы были собраны вручную, затем поделены на блоки с помощью программы, написанной на языке Python. Текст отзывов из блоков «Достоинства» добавлялся в корпус «Достоинства», текст отзывов из блоков «Недостатки» – в корпус «Недостатки», текст из блоков «Комментарий» в корпусы не добавлялся.

Впоследствии, два полученных корпуса были проанализированы с помощью инструмента “Sketch Engine”. Объём корпуса «Достоинства» составил 18,807 слов (25, 928 токенов), объём корпуса «Недостатки» – 25,763 слов (33, 442 токена).

Объём тестовой выборки составил 1000 отзывов. Данные отзывы были собраны автоматически, с помощью программы “Humpty-Dumpty”. Отзывы добавлялись в корпус в следующем виде: ссылка на отзыв – оценка продукта (от 1 до 5) – текст из блока «Достоинства» – текст из блока «Недостатки» – текст из блока «Комментарий».

Пример: «https://market.yandex.ru/product/12711629/reviews?hid=91013&CAT_ID=432460&show-old=1&page_num=1 4 гиговая видюха экран .2 usb ноут не оправдал ожидания иногда лагает не понятно и за чего . wot тянет на средних 20-30 fps. думаю вернуть»

Дальнейшее деление на блоки происходит по символу табуляции.

4.2. Составление словарей для инструмента анализа отзывов с помощью “Sketch Engine”

На следующем этапе, то есть в ходе работы программы, вывод о том, считает ли определённый пользователь какой-либо параметр продукта его достоинством или недостатком, делается в зависимости от того, найдёт ли программа в тексте отзыва слово или словосочетание из соответствующего словаря. В качестве примера рассмотрим параметр «Экран». Естественно, что не всегда пользователи выражают своё мнение однозначно, например, фразой «Мне нравится экран этого ноутбука». Он может выразить это и другими словами. Например, он может сказать, что у этого ноутбука «великолепные углы обзора» или «прекрасное разрешение». Всё это указывает на то, что пользователь считает экран данной модели её достоинством, но слово «экран» в отзыве не фигурирует. Следовательно, программа должна учитывать этот момент для получения более точных и качественных результатов. С этой целью был произведён анализ тренировочной выборки, в результате чего для каждого параметра был составлен словарь слов и словосочетаний, которые отсылают к данному параметру. В случае с параметром «Экран» в такой словарь входят такие слова и словосочетания, как «угол обзора», «разрешение», «full hd» и т.д.

Для составления словарей был использован инструмент “Sketch Engine” (https://the.sketchengine.co.uk), затем полученные с его помощью результаты были проанализированы и подкорректированы экспертами в области лингвистики и электроники.

В качестве примера рассмотрим процесс составления словарей на основании корпуса «Достоинства».

4.2.1. Извлечение ключевых слов и словосочетаний из корпуса. Функция “Keywords/terms”

Первым этапом стало использование функции “Keywords/terms” для выявления параметров, которые чаще всего встречаются в отзывах. Однако, это помогло выявить не только параметры, для которых в дальнейшем будут составлены словари (названия словарей), но и те слова и словосочетания, которые в них войдут, к примеру, синонимы заголовков или слова, которые имеют отношение к данному параметру. К примеру, наряду со словом «видеокарта», который станет заголовком словаря, в список ключевых слов для этого параметра были включены слова «дискретка», «geforce» и т.д., которые будут добавлены в файл словаря «Видеокарта».

Выделение ключевых слов и словосочетаний в Sketch Engine происходит по следующему принципу: программа считает, сколько раз определённое слово встретилось в исследуемом корпусе и сколько раз в справочном корпусе, затем полученные числа умножаются либо на тысячу, либо на миллион, чтобы предоставить информацию о частоте на тысячу или на миллион, а затем одно число делится на другое, чтобы получить их соотношение (Kilgarriff, 2009). Это соотношение является коэффициентом «терминологичности» данного слова, то есть указывает, насколько данное слово близко к понятию ключевого слова по отношению к исследуемому корпусу. В результате использования функции “Keywords/terms” мы получаем два списка – список ключевых слов типа “Single-word”, то есть состоящих из одного слова, и список ключевых слов типа “Multi-word”, то есть список терминологических словосочетаний. В полученных списках вышеупомянутый коэффициент обозначается словом “Score”. Формула, по которой считается параметр Score, выглядит следующим образом:

$$\frac{\int_{pmfocus} + N}{\int_{pmref} + N}$$

где $\int_{pmfocus}$ - это нормализованная частота (на миллион) слова в целевом корпусе, $\int_{pmref}$ - нормализованная частота (на миллион) слова в справочном

корпусе, а N – так называемый параметр сглаживания (значение по умолчанию равно 1).

Рассмотрим результаты применения данной функции на корпусе «Достоинства».

Были установлены следующие настройки:

Исследуемый корпус: Достоинства

Справочный корпус: Russian Web 2011 (ruTenTen11)

Параметр сглаживания N: 1 (При увеличении значения параметра слова с более высокой частотой добавляются в список ключевых слов)

Атрибут корпуса (атрибут корпуса, который используется для извлечения ключевых слов): lc

Минимальная частота: 1 (в исследуемом корпусе)

Максимальное количество ключевых слов: 100

Максимальное количество словосочетаний: 100

Справочный корпус для словосочетаний: Russian Web 2011 sample (ruTenTen11)

Релевантные слова типа Single-word (отобрано вручную из первоначального списка)

Score – коэффициент «терминологичности» данного слова

F – частота в исследуемом корпусе

RefF – частота в справочном корпусе

Single-word	Score	F	RefF
тачпад	1,167.89	43	7,694
клавиатура	870.18	118	77,348
греется	796.83	40	17,135
fullhd	524.57	16	3,259
видеокарта	486.50	32	28,132
оперативки	431.70	16	7,893
производительный	397.64	16	10,135
шустрый	391.13	17	12,410
оперативы	302.28	9	2,772

лёгкий	301.84	29	49,519
экран	276.15	166	405,600
fhd	259.26	7	826
клавиатуры	256.63	49	116,409
win	253.33	23	45,804
люфтов	250.16	8	4,340
оперативка	245.97	7	1,859
трекпад	220.72	6	968
тонкий	210.33	60	182,937
мультитач	199.43	7	6,558
батарея	195.37	28	82,861
тачпада	194.14	6	3,604
процессор	181.56	50	175,983
клавы	180.66	6	5,236
цветопередача	174.85	7	10,050
дискретной	174.02	7	10,185
сборка	170.79	38	138,700
скрипов	160.38	5	3,814
стильный	158.82	25	92,819
клава	153.20	7	14,053
легкий	152.87	78	341,592
тач	150.46	5	5,270
люфтит	142.99	4	1,570
маркий	139.23	4	2,106
retina	135.97	4	2,595
ddr	134.64	10	34,221
geforce	131.58	11	40,798
зарядка	127.86	13	53,547
батарейка	127.34	6	15,084
винды	123.43	6	16,141
фпс	122.54	6	16,389
шустрая	121.05	4	5,168
клавиш	118.94	18	88,571
шустро	116.85	5	12,044
дискретка	116.46	3	39

Релевантные слова типа Multi-word (отобрано вручную из первоначального списка)

Multi-word	Score	F	RefF
жёсткий диск	695.23	18	0
лучший звук	463.82	12	0
угол обзора	339.57	26	2,453
алюминиевый корпус	183.34	7	600
система охлаждения	158.58	23	5,769
красивейший дизайн	155.27	4	0
приятный материал	150.75	4	38
матовый покрытие	116.70	3	0
высшее разрешение	116.70	3	0
лучшая цена	116.70	3	0
мощное железо	113.75	3	33
оперативная память	116.53	20	7,057
максимальная яркость	100.44	3	204
заряд батареи	96.92	4	756
ценовой категория	80.03	6	2,388
скорость работы	78.54	7	3,073
качество материалов	75.78	4	1,316
внешний вид	37.33	34	42,830
блок питания	22.41	4	7,434
операционная система	11.89	5	19,186
тишайший вентилятор	78.14	2	0
разумнейший деньга	78.14	2	0
пошире диапазон	78.14	2	0
неплохой динамика	78.14	2	1

Распределение ключевых слов по словарям.

1. Тачпад: тачпад, трекпад, мультитач, тачпада, тач
2. Клавиатура: клавиатура, клавиатуры, клавы, клава, клавиш
3. Система охлаждения: не греется, система охлаждения, вентилятор
4. Экран: fullhd, экран, fhd, цветопередача, retina, угол обзора, разрешение, яркость

5. Видеокарта: видеокарта, дискретной, geforce, фпс, дискретка
6. Производительность и скорость работы: производительный, шустрый, шустрая, шустро, скорость работы
7. Оперативная память: оперативки, оперативы, оперативка, ddr, оперативная память
8. Габариты: лёгкий, тонкий, легкий
9. Операционная система: win, винды, операционная система
10. Корпус: люфтов, скрипов, сборка, люфтит, маркий, корпус, материал, матовое покрытие, материалов
11. Батарея: батарея, зарядка, батарейка, заряд, батареи, блок питания
12. Процессор: процессор, железо
13. Внешний вид: стильный, дизайн, внешний вид
14. Жёсткий диск: жёсткий диск
15. Цена: ценовой категории, разумные деньги
16. Звук: динамики, звук

С целью экономии места в словарях и соответственно с целью ускорения работы программы в будущем, некоторые словосочетания (из списка Multi-word) в словари добавляются не полностью, а только одним словом, так как даже если в тексте отзыва встречается только одно это слово, оно всё равно указывает на упоминание данного параметра. К примеру, в корпусе часто встречаются словосочетания типа «неплохой звук», «прекрасный звук», «лучший звук» и т.д. В таком случае, в словарь будет добавлено только слово «звук», так как любое из этих словосочетаний, как и просто упоминание параметра «звук» в блоке «Достоинства» говорит о том, что пользователь считает звук устройства его положительным качеством.

Устойчивые словосочетания были добавлены полностью, так как части таких словосочетаний вне данного словосочетания приобретают другое значение (пример: жёсткий диск, внешний вид).

4.2.2. Составление тезаурусов для каждого параметра. Функция “Thesaurus”

Тезаурус создаётся следующим образом: исследуется корпус; производится поиск контекстов для каждого слова; производится поиск слов, которые встречаются в одинаковом контексте. Для каждого слова, слова, которые разделяют с этим словом наибольшее количество контекстов (в соответствии со статистикой, которая также учитывает частоту их встречаемости), являются его ближайшими соседями (Rychlý and Kilgarriff, 2007).

Алгоритм работает так: если при построении тезауруса обнаруживаются такие примеры, как «пить кофе» и «пить чай», это может свидетельствовать о том, что слова «кофе» и «чай» похожи. Можно сказать, что они имеют общий коллокат «пить» (глагол), в связи типа “OBJECT-OF” (Kilgarriff, Baisa, Bušta, Jakubícek, Kovář, Michelfeit, Rychlý and Suchomel, 2014).

Функция «Тезаурус» применялась к каждому параметру продукта, отобранному на предыдущем этапе анализа корпусов. В качестве таких параметров были выделены: тачпад, клавиатура, система охлаждения, экран, видеокарта, производительность и скорость работы, оперативная память, габариты, операционная система, корпус, батарея, процессор, внешний вид, жёсткий диск, цена, звук.

В качестве примера того, как эта применялась для составления словарей, рассмотрим процесс составления тезауруса для параметра «Батарея» и его последующую корректировку.

При использовании этой функции были установлены следующие настройки:

Лемма: видеокарта

Часть речи: существительное

Максимальное количество элементов: 60

Минимальное значение параметра “Score”: 0.0

Минимальное значение меры сходства между элементами кластера: 0.15

Тезаурус для параметра «Батарея» на материале корпуса «Достоинства»

Lemma – лемма

Score – коэффициент сходства

Freq – частота встречаемости слова в исследуемом корпусе

Lemma	Score	Freq
аккумулятор	0.738	19
заряд	0.489	21
зарядка	0.256	29
ноутбук	0.088	132

Из полученного списка в словарь параметра «Батарея» было добавлено только слово «аккумулятор», так как слова «заряд» и «зарядка» были добавлены в него на предыдущем этапе анализа, а слово «ноутбук» не является синонимом слова «батарея» и его упоминание в тексте отзыва не поможет программе сделать вывод о параметре «Батарея».

В связи с размером исследуемого корпуса для некоторых параметров система не смогла составить тезаурусы. В случае, когда они всё же были составлены, полученные данные лишь незначительно обогатили созданные словари. Впоследствии, синонимы и слова, близкие по смыслу к теме словарей, были добавлены в них вручную, тем не менее, использование тезаурусов помогло избежать упущения некоторых элементов, релевантных для составляемых словарей.

В будущем планируется увеличить объём корпусов с целью составления более подробных тезаурусов и добавления новых элементов в словари.

4.2.3. Извлечение слов, которые часто встречаются вместе с исследуемыми словами. Функции “Word Sketch” и “Sketch diff”

“Word Sketch” представляет собой краткое описание грамматического поведения слова и коллокаций, в которых оно встречается, на одной странице. В этом кратком описании перечисляются леммы, которые встречаются вместе с целевым словом. Эти леммы разделяются на группы в зависимости от того, в каких отношениях они находятся с исследуемым словом (Kilgarriff, Rychlý, Kovář and Baisa, 2012).

«Ворд-скетчи» можно рассматривать как устойчивые словосочетания, содержание которых, с одной стороны, обусловлено синтаксисом, ограничивающим формирование словосочетаний в данном языке, а с другой стороны, вероятностью, которая тесно связана с семантикой и/или словоупотреблением (Khokhlova, Zakharov, 2010).

При работе с функцией “Sketch diff” сравниваются «ворд-скетчи» двух слов. Полученные результаты также сортируются в зависимости от того, в каких отношениях отобранные леммы находятся с исследуемыми словами. В итоге, пользователь получает несколько списков. В заголовке каждого списка фигурирует название типа отношений между полученными леммами и исследуемыми словами. К примеру, отношения типа «и/или» (добрый или злой), «subject_of» (человек идёт), «inst_modifies» (обливается водой) и т.д. В самом списке формируются три блока: зелёный – леммы, которые встречаются в контексте только с первым исследуемым словом или чаще всего вместе с ним; белый – леммы, которые встречаются примерно в одинаковом объёме и с первым и со вторым словом; красный – леммы, которые встречаются только вместе со вторым словом или чаще всего вместе с ним.

Для экономии времени в данном исследовании чаще использовалась именно функция “Sketch diff”, так как она позволяет анализировать «ворд-скетчи» сразу для двух слов. Анализ проводился на материале корпуса Russian Web 2011 (ruTenTen11), так как использование этого корпуса позволило добавить в словари большее количество слов и учесть те коллокаты, которые не распространены в корпусах, составленных на базе отзывов, но которые также имеют отношение к рассматриваемым параметрам.

После формирования таких списков для уже добавленных в словари слов были получены другие слова, тесно связанные с темами словарей. В качестве примера рассмотрим часть списка “Sketch diff”, составленного для слов «батарея» и «аккумулятор».

Список “Sketch diff” для слов «батарея» и «аккумулятор» (типы отношений «и/или» и «subject_of»)

Второй столбец – общая частота встречаемости с первым словом

Третий столбец – общая частота встречаемости со вторым словом

Четвёртый столбец – коэффициент схожести относительно первого слова

Пятый столбец – коэффициент схожести относительно второго слова

и/или	**39,339**	**33,992**	**0.07**	**0.08**
ветряк	160	0	6.8	--
эскадрон	227	0	6.8	--
дивизион	557	0	6.6	--
стояк	232	0	6.5	--
ветрогенераторы	104	0	6.4	--
обогреватель	271	0	6.2	--
рота	459	0	6.0	--
взвод	294	0	5.8	--
ветряков	63	0	5.6	--
ветрогенераторов	59	0	5.5	--
подоконник	226	0	5.4	--
радиатор	664	49	5.8	2.1
аккумулятор	2,064	196	7.3	3.9
инвертор	99	102	5.6	5.7
электромотор	152	182	6.3	6.6

генератор	406	877	4.6	5.7
инверторы	20	51	3.9	5.4
стартер	92	390	5.0	7.1
атомайзера	12	51	3.3	5.6
зарядник	11	56	3.1	5.6
батарея	330	2,063	4.1	6.7
батарейка	17	2,258	2.0	9.1
автошины	0	61	--	5.7
автошин	0	65	--	5.8
атомайзер	0	66	--	5.9
subject_of	**47,689**	**42,372**	**0.08**	**0.11**
греть	467	0	6.9	--
село	539	0	6.5	--
топить	86	0	4.9	--
потечь	94	0	4.9	--
садиться	660	682	5.4	5.5
подзаряжаются	45	50	4.9	5.2
прослужить	82	104	4.6	5.0
заряжаться	962	1,218	8.7	9.2
разряжаться	594	824	8.4	9.0
подзаряжается	79	135	5.7	6.7
разрядиться	570	1,086	8.4	9.5
зарядиться	109	234	6.1	7.3
сдохнуть	145	367	6.1	7.5
отключить	31	112	3.3	5.2
сесть	382	2,052	4.8	7.2
подзарядить	14	69	3.2	5.6
подсесть	16	85	2.9	5.4
заряжать	79	626	4.6	7.6
подзаряжает	10	111	2.8	6.4
разряжать	12	255	2.4	6.9
отсоединить	0	41	--	4.9
сажать	0	202	--	5.2
Заряжается	0	58	--	5.5
разрядить	0	108	--	6.2
зарядить	0	250	--	6.2

В этой главе рассмотрены только два вида отношений, которые были выделены для исследуемых слов. В процессе анализа корпусов с помощью этой функции были также получены элементы, связанные с исследуемыми словами другими типами отношений, однако, здесь они приводиться не будут, так как все полученные леммы из других списков были признаны нерелевантными.

В данном исследовании не имеет большого значения, к какому из исследуемых слов относится та или иная лемма, поэтому цвет блоков не учитывался. Имеет значение только то, относится ли рассматриваемая лемма к теме словаря.

Впоследствии из полученного списка были отобраны и учтены при составлении словарей слова, которые имеют отношение к теме исследуемого материала, то есть к ноутбукам, и к параметру, для которого был составлен список, то есть, в данном случае к параметру «Батарея». Были отобраны следующие слова: зарядник, подзаряжаются, заряжаться, разряжаться, подзаряжается, разрядиться, зарядиться, подзарядить, заряжать, подзаряжает, разряжать, заряжается, разрядить, зарядить.

Затем, на основании отобранных элементов были выделены части, общие для нескольких слов, чаще всего относящихся к одной части речи, и добавлены в словари. Словарь для параметра «Батарея», полученный после трех этапов исследования, представлен ниже:

заряд
заряжают
заряжать
заряжен
заряжая
заряжает
батаре
аккум

блок питан
блоке питан
блоком питан
блока питан
блоки питан
разряжается
разряжался
разряжалась
разряжались
разряжен
разряди
разряжая

В результате, программа будет искать в отзывах не слова, а вышеперечисленные части слов. Это было сделано для того, чтобы не перегружать словари формами слов и обеспечить более быструю работу программы. Однако некоторые элементы были добавлены целиком, чтобы избежать в результатах однокоренных слов, не имеющих отношения к анализируемому параметру.

Результаты анализа корпусов «Достоинства» и «Недостатки» в системе “Sketch Engine” описаны в Приложении.

4.2.4. Проверка релевантности элементов словаря с помощью конкорданса. Функция “Concordance”

Для того чтобы убедиться, что при дальнейшей работе программы будут делаться правильные выводы относительно мнения пользователя о том или ином параметре, все полученные элементы словарей были проверены с помощью функции “Concordance”. Было необходимо проверить, что все элементы словарей указывают именно на упоминание определённого параметра, а не отсылают к каким-либо другим деталям анализируемого объекта или к явлениям, не имеющим отношения к сфере рекомендации. К примеру, важно получить подтверждение тому, что упоминание слова «батарея» всегда (или почти всегда) отсылает именно к детали ноутбука, а не подразумевает военное подразделение.

Конкорданс, то есть список употреблений какого-либо слова в определённом окружении, это основа использования корпусов и очень важная часть лексикографического анализа. Строки конкорданса могут быть представлены в формате предложения или в формате KWIC (“Key Word in Context” - Ключевое Слово в Контексте). При использовании формата KWIC, предпочитаемого в лексикографии, система демонстрирует строку с контекстом для каждого употребления слова с этим словом посередине. Используя функцию составления конкорданса, лексикографы могут просмотреть данные и быстро получить представление о моделях словоупотребления, семантике, составных словах и т.д. (Kilgarriff and Kosem, 2012).

В случае если на предшествующих этапах какой-либо элемент словаря был добавлен ошибочно, то есть употребляется зачастую не в том значении, которое отсылает к заголовку словаря, этот элемент из него исключается.

4.2.5. Использование вышеперечисленных функций на материале корпуса «Недостатки»

После анализа корпуса «Достоинства» те же самые функции были применены и к корпусу «Недостатки». В случае, если при анализе второго корпуса были выделены какие-либо релевантные слова или словосочетания, которые не были обнаружены на этапе анализа корпуса «Достоинства», они были добавлены в словари.

Словари для всех отобранных параметров можно найти в Приложении 1.

4.3. Разработка программы анализа отзывов

В процессе данного исследования был разработан механизм автоматического анализа отзывов, который на основании словарей, составленных на предшествующих этапах, и на основании пользовательских отзывов формирует для пользователей рекомендации. Рекомендации представляют собой оценки по 5-тибальной шкале каждому параметру оцениваемого объекта. Для исследуемых объектов было выявлено 16 основных параметров: тачпад, клавиатура, система охлаждения, экран, видеокарта, производительность и скорость работы, оперативная память, габариты, операционная система, корпус, батарея, процессор, внешний вид, жёсткий диск, цена, звук. Следовательно, каждому продукту на основании анализируемых отзывов система выставляет 16 оценок.

Алгоритм работы программы:

1) На вход программа получает корпус отзывов. Каждый отзыв в корпусе представлен в следующем виде:

«ссылка на отзыв» + оценка продукта + блок «Достоинства» + блок «Недостатки» + блок «Комментарий»

Пример (стрелкой обозначается символ табуляции):

«https://market.yandex.ru/product/12711629/reviews?hid=91013&CAT_ID=432460&show-old=1&page_num=1→1→ Отсутствуют полностью→ Плохое качество сборки. Плохой дешевый пластик. При перевороте ноутбука, перестает определяться жесткий диск. Лечится еще одним переворотом. →Ноутбук приносили на настройку. Все что плохое может

быть в ноутбуках, в этом есть: плохое качество, слабое железо и брак. Никому не рекомендую к покупке».

Кроме корпуса отзывов на вход программа получает словари для каждого параметра продукта. На предыдущем этапе исследования было отобрано 16 основных параметров, чаще всего упоминающихся в отзывах. Соответственно, было составлено 16 словарей. Подробное описание процесса составления словарей можно найти в предыдущем разделе этой работы. В результате исследования было принято решение добавлять элементы в словари не всегда в виде целых слов, иногда опуская окончания, так как такие элементы отсылают к любой форме подразумеваемого слова, при этом помогая уменьшить количество времени, требуемое программе для анализа отзывов.

К примеру, словарь для параметра «Батарея» включает следующие элементы: заряд, заряжают, заряжать, заряжен, заряжая, заряжает, батарее, аккум, блок питан, блоке питан, блоком питан, блока питан, блоки питан, разряжается, разряжался, разряжалась, разряжались, разряжен, разряди, разряжая.

2) В первом фрагменте отзыва, то есть в ссылке на отзыв после фрагмента “product/” указан номер продукта. Сначала программа сортирует отзывы по этому номеру и добавляет все отзывы с одинаковым номером в отдельный список.

3) Затем каждый отзыв разделяется на несколько блоков: ссылка, оценка, достоинства, недостатки, комментарий. Разделение на блоки происходит по знаку табуляции.

4) Для каждого продукта создаётся определённое количество переменных, каждая из которых представляет собой оценку тому или иному параметру продукта. Изначально каждому параметру присваивается значение 0.

5) Затем система приступает к выставлению оценки по каждому параметру. Сначала формируется оценка для первого параметра для

каждого из продуктов, потом программа переходит к оценке второго параметра, и т.д. Рассмотрим этот этап более подробно.

5.1) Программа открывает файл со словарём для первого параметра, в нашем случае это параметр «Цена». Затем она открывает список отзывов для первого продукта и в каждом отзыве сравнивает содержание блока «Достоинства» с содержанием словаря параметра «Цена». Происходит подсчёт совпадений для каждого отзыва.

5.2) Происходит сравнение содержания блока «Недостатки» для анализируемого отзыва с содержанием словаря параметра «Цена» и подсчёт совпадений.

5.3) Сравнивается количество совпадений, найденных в пункте 5.1 и в пункте 5.2. В случае если количество совпадений, найденных на этапе 5.1 больше количества совпадений, найденных на этапе 5.2, система делает вывод, что автор анализируемого отзыва считает цену данного продукта его достоинством, и увеличивает значение параметра «Цена-Достоинство» для данного продукта. В случае если количество совпадений, найденных на этапе 5.2, больше количества совпадений, найденных на этапе 5.1, система увеличивает значение параметра «Цена-Недостаток». В случае равенства увеличиваются значения обоих параметров (кроме случаев, когда количество совпадений, найденных на обоих этапах, равно нулю).

5.4) Проанализировав все отзывы, система получает два значения для каждого параметра, то есть количество положительных оценок и количество отрицательных оценок. Затем высчитывается процент положительных оценок по формуле:

$$\frac{кол\text{-}во\ полож.\ оценок}{кол\text{-}во\ полож.оценок + кол\text{-}во\ отр.оценок} * 100$$

5.5) Так как оценка для всего продукта на сайте «Яндекс Маркет» выставляется по 5-тибальной шкале, где 1 – «ужасная модель», 5 – «отличная модель», оценки для отдельных параметров продуктов приводятся к такому же формату. Если процент положительных оценок находится в диапазоне от 0% до 20%, анализируемому параметру присваивается оценка 1, от 20% до 40% -

оценка 2, от 40% до 60% - оценка 3, от 60% до 80% - оценка 4, от 80% до 100% - оценка 5.

В том случае, если для выставления оценки по данному параметру недостаточно данных, то есть и количество положительных оценок и количество отрицательных оценок равно нулю, параметру присваивается значение «нет отзывов».

6) После сравнения содержания отзывов со словарём первого параметра, программа переходит к сравнению содержания отзывов со словарём следующего параметра.

7) Полученные оценки сохраняются в файл вместе с соответствующими им номерами продуктов. На выходе получаем список следующего вида:

Номер продукта: 10781899; Цена: 1; Внешний вид: 5; Звук: 2; Экран: 5; Батарея: 3; Тачпад: 1; Скорость и производительность: 5; Клавиатура: 4; Система охлаждения: 5; Видеокарта: нет отзывов; Оперативная память: нет отзывов; Габариты: 3; Операционная система: 1; Корпус: 5; Процессор: нет отзывов; Жёсткий диск: 3

Фрагменты программы с пояснениями можно найти в Приложении 2.

4.4. Оценка результатов. Правильность, точность, полнота

После выполнения алгоритма и сохранения полученных результатов в файл, был проведён анализ тестовой выборки вручную, то есть с точки зрения пользователей.

Затем были сравнены результаты, полученные после обработки выборки программой, и результаты, полученные после анализа, проведённого вручную.

На основании этого сравнения для каждого параметра были подсчитаны следующие величины:

TP (true-positive) – истинно-положительные решения (продукты, для которых совпала положительная оценка пользователем и системой)

FP (false-positive) – ложно-положительные решения (продукты, которые система порекомендовала, но которые не понравились пользователю)

FN (false-negative) – ложно-отрицательные решения (продукты, которые система не порекомендовала, но которые понравились пользователю)

Рекомендацией считается оценка равная 3 или выше 3, параметры с оценкой ниже 3 считаются не рекомендованными.

Всего в тестовой выборке было рассмотрено 29 продуктов, поэтому максимальное число совпадений для какого-либо параметра – 29.

Параметр	**tp**	**fp**	**fn**
цена	23	3	2
внешний вид	29	6	5
звук	18	0	2
экран	17	4	2
батарея	21	1	3
тачпад	22	2	1

скорость	19	2	6
клавиатура	19	1	4
система охлаждения	23	1	0
видеокарта	24	2	1
оперативная память	23	1	2
габариты	25	2	2
операционная система	15	5	3
корпус	17	5	3
процессор	23	1	3
жёсткий диск	17	1	5

На основании этих величин для каждого параметра были высчитаны меры правильности, точности и полноты по следующим формулам:

1. Accuracy (правильность): $\frac{P}{N}$,

   где P – количество совпадений оценок пользователя с оценками системы, а N – общее количество продуктов

2. Precision (точность): $\frac{TP}{TP + FP}$

3. Recall (полнота): $\frac{TP}{TP + FN}$

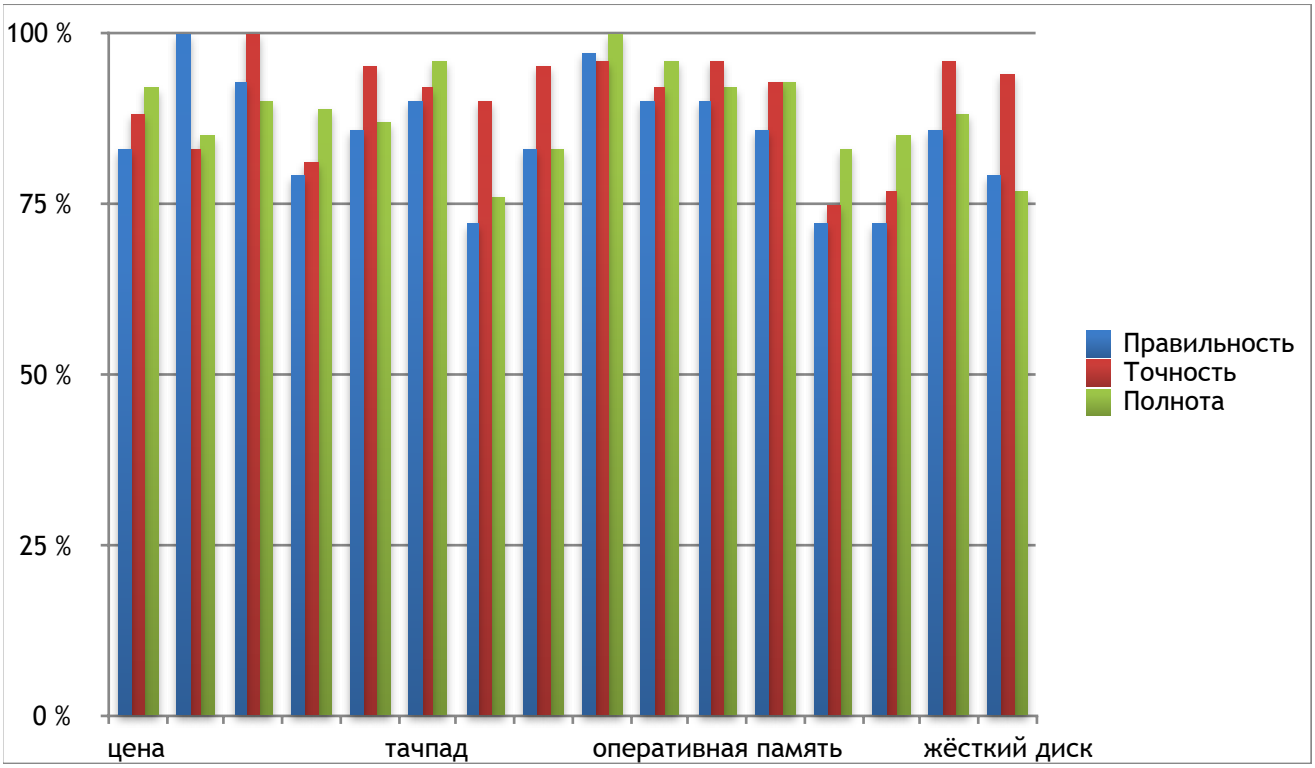

Рисунок 1. Меры правильности, точности и полноты для параметров продуктов

Параметр	Правильность	Точность	Полнота
цена	83%	88%	92%
внешний вид	100%	83%	85%
звук	93%	100%	90%
экран	79%	81%	89%
батарея	86%	95%	87%
тачпад	90%	92%	96%
скорость	72%	90%	76%
клавиатура	83%	95%	83%
система охлаждения	97%	96%	100%
видеокарта	90%	92%	96%
оперативная память	90%	96%	92%
габариты	86%	93%	93%
операционная система	72%	75%	83%
корпус	72%	77%	85%
процессор	86%	96%	88%
жёсткий диск	79%	94%	77%

Как видно на графике, результаты, полученные в ходе работы программы, достаточно близки к пользовательским предпочтениям. Больше всего ошибок было допущено при анализе следующих параметров:

1. Экран (низкие правильность и точность).
2. Скорость (низкие правильность и полнота).
3. Операционная система (все 3 показателя сравнительно низкие).
4. Корпус (все 3 показателя).
5. Жёсткий диск (низкие правильность и полнота).

Зачастую ошибки связаны с тем, что элемент анализируемого словаря встречается в тексте отзыва в составе какого-либо словосочетания, которое не является устойчивым, и упоминание которого в действительности не отсылает к исследуемому параметру, а отсылает к чему-либо с ним связанному.

К примеру, словосочетание «жёсткий диск» может употребляться в составе более распространённого словосочетания, такого как «слот для жёсткого диска». В таком случае, если пользователь напишет такой отзыв, как «хороший слот для жёсткого диска», это не означает, что сам жёсткий диск ему нравится, однако, система сделает другой вывод.

Для решения этой проблемы в будущих исследованиях мы также уделим внимание часто встречающимся в тексте отзывов словосочетаниям, состоящим из 3 и более элементов.

В случае с параметром «Операционная система» ошибки возникают в связи с употреблением словосочетаний типа «невозможно поставить виндоус». Не обязательно, что автор такого комментария считает операционную систему ноутбука его недостатком, однако система находит в отзыве слово «виндоус» и делает противоположный вывод.

Конечно, частота подобного рода ошибок сравнительно не велика, но для лучшего качества рекомендаций, нужно попробовать свести их количество к минимуму. Для этого в последующих работах будет проанализировано окружение, в которых чаще всего возникают подобные ошибки, и по возможности учтено при последующем анализе отзывов.

Заключение

В ходе данного исследования были выполнены следующие задачи:

1. Проведён анализ проблем, возникающих при работе рекомендательных систем, предложены пути их решения с помощью использования инструмента автоматического анализа отзывов.

2. Проведён сравнительный анализ существующих алгоритмов рекомендаций, изучены основные преимущества и недостатки существующих РС.

3. На основании рассмотренной таксономии разработана методика алгоритма рекомендаций, основанного на пользовательских отзывах.

4. С учётом разработанной методики составлены словари, необходимые для автоматического анализа отзывов.

5. Разработана программа автоматического анализа отзывов.

6. Проведён анализ собранных отзывов с помощью разработанного инструмента.

7. Произведена оценка полученных результатов. Предложены пути решения возникших в ходе работы осложнений.

8. Рассмотрены возможности дальнейшего усовершенствования алгоритма.

В результате данного исследования была разработана программа для автоматического анализа отзывов в рекомендательных системах. Проанализировав результаты работы программы, мы пришли к выводу, что данный инструмент отличается очень хорошими показателями (на основании мер, расчитанных в разделе 4.4.) и предоставляет рекомендации, максимально приближенные к реальным предпочтениям пользователей. Отметим также, что система составляет рекомендации на основании отзывов других пользователей, за счёт чего может обеспечить увеличение уровня доверия к ресурсу и к предоставляемым рекомендациям.

Библиография

1. Глибовец Н.Н., Сидоренко М.О. “Создание рекомендационной системы учебного типа с использованием фреймворка Windows Communication Foundation” // Problems of Computer Intellectualization, с. 176-181, 2012.

2. Гомзин А.Г., Коршунов А.В. “Системы рекомендаций: обзор современных подходов” // Труды Института системного программирования РАН, т. 22, с. 401–417, 2012.

3. Дьяконов А.Г. “Алгоритмы для рекомендательной системы: технология LENKOR” // Бизнес-Информатика, Т. 1, № 19. – С. 32-39, 2012.

4. Нефедова Ю. С.. “Архитектура гибридной рекомендательной системы GEFEST (Generation–Expansion–Filtering–Sorting–Truncation)”, Системы и средства ин-форм., 2012, том 22, выпуск 2, с. 176–196, 2012.

5. Пятикоп Е.Е. “Использование сингулярного разложения матриц в коллаборативной фильтрации” // Проблеми інформатизації та управління, 4(44), с. 76-81, Национальный авиационный университет, Киев, 2013.
6. Пятикоп Е.Е. “Исследование метода коллаборативной фильтрации на основе сходства элементов” // Наукові праці Донецького національного технічного університету серія: "Інформатика, кібернетика та обчислювальна техніка", №2, с. 109-114, 2013.
7. Ролгин Р.И. “ Метрики оценки качества работы систем коллаборативной фильтрации”. Материалы Международной конференции и молодёжной школы «Информационные технологии и нанотехнологии», с. 1092-1095, Издательство СГАУ, 2016.
8. Савкова Е.О., Привалов М.В., Чепикова Е.Д. “Исследование алгоритмов рекомендательных систем” // Информатика и кибернетика 2 (4), с. 57-60, 2016.
9. Федоровский А. Н., Логачева В. К. “Архитектура рекомендательной системы, работающей на основе неявных пользовательских оценок” // Электронные библиотеки: перспективные методы и технологии, электронные коллекции // Труды XIII Всероссийской научной конференции RCDL’2011. — Воронеж: Воронежский госуниверситет, с. 76–82, 2011.
10. Adomavicius G. and Kwon Y. “New recommendation techniques for multicriteria rating systems” in IEEE Intelligent Systems, 22, p. 48–55, May 2007.
11. Adomavicius G. and Tuzhilin A. “Context-aware recommender systems” in F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor (Eds), “Recommender Systems Handbook”, Springer, pp. 217–253, 2011.
12. Adomavicius G., Tuzhilin A. “Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions”. In: IEEE Transactions of Knowledge and Data Engineering, 2005.
13. Aggarwal C.C. “Recommender Systems: The Textbook”, 498 pages, p.168, Springer, March 2016.

14. Agrawal R. and Srikant R. “Fast algorithms for mining association rule” in Proceedings of the 20th International Conference on Very Large Databases, pp. 487-499, 1994.
15. Bendakir N. and Aïmeur E. “Using association rules for course recommendation” in Proceedings of the AAAI Workshop on Educational Data Mining, pp. 31-40, July 2006.
16. Bennett J., Lanning S. “The Netflix Prize”. In Proc. of KDD Cup Workshop at SIGKDD-07, 13th ACMInt. Conf. on Knowledge Discovery and Data Mining, 2007.
17. Burke R. “Hybrid Recommender Systems: Survey and Experiments” in UMUAI 12 (4), pp. 331- 370, 2002.
18. Chen L., Chen G. and Wang F. “Recommender systems based on user reviews: the state of the art”. User Modeling and User-Adapted Interaction, Volume 25, Issue 2, pp. 99–154, June 2015.
19. Dehuri S. “Intelligent Techniques in Recommendation Systems: Contextual Advancements and New Methods”, IGI Global, 2012, p. 51.
20. Desrosiers C., Karypis G. “A comprehensive survey of neighborhood-based recommendation methods” in: F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), “Recommender Systems Handbook”, Springer, pp. 107–144, 2011.
21. He X., Chen T., Kan M.-Y., and Chen X. “Tri-rank: Review-aware explainable recommendation by modeling aspects” in Proceedings of the 24th ACM Conference on Information and Knowledge Management, CIKM ’15, Melbourne, Australia, 2015.
22. Hooda R., Singh K. and Dhawan S. “A Study of Recommender Systems on Social Networks and Content-based Web Systems” in *International Journal of Computer Applications* 97(4), pp. 23-28, July 2014.
23. Hu Y., Koren Y. and Volinsky C. “Collaborative Filtering for Implicit Feedback Datasets”. Data Mining, ICDM'08. Eighth IEEE International Conference on, pp. 263-272, 2008.

24. Jain A., Jain V. and Kapoor N. “A literature survey on recommendation systems based on sentimental analysis” in Advanced Computational Intelligence: An International Journal (ACII), Vol.3, No.1, pp. 25-36, January 2016.

25. Jakob N., Weber S. H., Müller M. C., and Gurevych I. “Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations” in Proc. TSA ’09, pages 57–64, New York, NY, USA, ACM, 2009.

26. Khokhlova M. and Zakharov V. “Studying Word Sketches for Russian” in Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC’12) Malta, pp. 3491–3494, 2010.

27. Khusro S., Ali Z., Ullah I. “Recommender Systems: Issues, Challenges, and Research Opportunities” in Information Science and Applications (ICISA), LNEE 376, pp. 1179-1189, 2016.

28. Kilgarriff A., Baisa V., Bušta J., Jakubícek M., Kovář V., Michelfeit J., Rychlý P., Suchomel V. “The Sketch Engine: ten years on” in *Lexicography* 1(1), pp. 7–36, 2014. DOI: 10.1007/s40607-014-0009-9. ISSN 2197-4292

29. Kilgarriff A. and Kosem I. “Corpus Tools for Lexicographers” in Electronic Lexicography. S. Granger & M. Paquot (eds). Oxford University Press, 2012.

30. Kilgarriff A., Rychlý P., Kovář V. and Baisa V. “Finding Multiwords of More Than Two Words” in Proceedings of the 15th EURALEX International Congress, Norway, pp. 693–700, 2012.

31. Kilgarriff A. “Simple maths for keywords” in *Proceedings of Corpus Linguistics Conference CL2009*, Mahlberg, M., González-Díaz, V. & Smith, C. (eds.), University of Liverpool, UK, July 2009.

32. Kulkarni K., Wagh K., Badgujar S., Patil J. “A Study Of Recommender Systems With Hybrid Collaborative Filtering” in International Research Journal of Engineering and Technology (IRJET), Volume: 03 Issue: 04, April 2016.

33. Lei K., Qin D., Zhang K. “A Trust-based Recommendation Model Constructed from Improved Page Rank Algorithm in a P2P Social Network”, CNSCE 2014, Shenzhen, P.R. China, pp. 335-340, Feb 2014.

34. Lenzini G., van Houten Y., Huijsen W., Melenhorst M. “Shall I trust a recommendation? Towards an evaluation of the trustworthiness of recommender sites” in Proceedings of the 13th East European conference on Advances in Databases and Information Systems, pp. 121-128, September 07-10, Riga, Latvia, 2009.

35. Levinas C.A. “An Analysis of Memory Based Collaborative Filtering Recommender Systems with Improvement Proposals”, Master of Science Thesis, 2014.

36. Liu C. L., Hsaio W. H., Lee C. H., Lu G. C. and Jou E. "Movie Rating and Review Summarization in Mobile Environment" in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 3, pp. 397-407, May 2012.

37. Marinho L.B., Nanopoulos A., Schmidt-Thieme L., Jäschke R., Hotho A., Stumme G., Symeonidis P. “Social tagging recommender systems” in: Ricci, F., Rokach, L., Shapira, B., Kantor, P.B. (eds.) “Recommender Systems Handbook”, pp. 615–644. Springer, Ljubljana, 2011.

38. Melville P., Mooney R. J., and Nagarajan R. “Content-boosted collaborative filtering for improved recommendations”. Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI- 02), pp. 187–192, Edmonton, Alberta, 2002.

39. Neumann A.W. “Recommender Systems for Information Providers: Designing Customer Centric Paths to Information”. Physica-Verlag, Heidelberg, p. 24, 2009.

40. Pathak D., Matharia S. and Murthy C. N. S. "ORBIT: Hybrid movie recommendation engine," Emerging Trends in Computing, Communication and Nanotechnology (ICE-CCN), pp. 19-24, International Conference on, Tirunelveli, pp. 19-24, 2013.

41. Pereira N. and Varma S.K. “Survey on Content Based Recommendation System” in International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 7 (1), pp. 281-284, 2016.

42. Poirier D., Tellier I., Fessant R., and Schluth J. “Towards text-based recommendations” in Proc. RIAO ’10, pp. 136–137, Paris, France, France, 2010.
43. Poriya A., Bhagat T, Patel N. and Sharma R. “Non-Personalized Recommender Systems and User-based Collaborative Recommender Systems” in International Journal of Applied Information Systems 6(9), pp. 22-27, March 2014.
44. Ricci F., Rokach L., Shapir B.. “Recommender Systems Handbook; Second Edition”. Springer, 2015.
45. Rychlý P. and Kilgarriff A. “An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments)” in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. Czech Republic, pp. 41–44, June 2007.
46. Schafer B., Konstan J. and Riedl J., "Recommender systems in e-commerce". Proceedings of the 1st ACM conference on Electronic commerce, New York City, pp. 158-166, 1999.
47. Shih Y. and Liu D. “Hybrid recommendation approaches: collaborative filtering via valuable content information” in 38th Hawaii International Conference on System Sciences (HICSS05), page 217b, Hawaii, USA, 1999.
48. Soni K., Goyal R., Vadera B. and More S. “A Three Way Hybrid Movie Recommendation System” in International Journal of Computer Applications (0975 – 8887), Volume 160 – No 9, February 2017.
49. Su X. and Khoshgoftaar T.M., “A Survey of Collaborative Filtering Techniques” in Advances in Artificial Intelligence, vol. 2009, Article ID 421425, 19 pages, 2009.
50. Wang F. and Chen L. “Recommendation based on mining product reviews’ preference similarity network” in The 6th Workshop on Social Network Mining and Analysis, 2012 ACM SIGKDD Conference on Knowledge Discovery and Data Mining, SNAKDD 2012, 2012.
51. Zhang B., Wang N., Jin H. “Privacy concerns in online recommender systems: Influences of control and user data input” in Symposium on Usable Privacy and Security (SOUPS), pp. 159-173, Menlo Park, CA, July 9-11, 2014.

52. Zhang W., Ding G., Chen L., Li C., and Zhang C. “Generating virtual ratings from chinese reviews to fuse into collaborating filtering algorithms”. ACM TIST, 2012.

53. Zhang XL, Lee TMD, Pitsilis G. “Securing recommender systems against shilling attacks using social-based clustering” in JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 28(4), pp. 616–624, July 2013.

54. Zhao S., Du N., Nauerz A., Zhang X., Yuan Q., Fu R. “Improved recommendation based on collaborative tagging behaviors” in Proceedings of the 13th International Conference on Intelligent User Interfaces, Gran Canaria, Canary Islands, Spain, ACM, IUI’08, pp. 413–416, 2008.

Словари

1. Петров Ю. А., Петрова Г. И. «Терминологический словарь-справочник: экономика, маркетинг, менеджмент. А – М», Издательские решения, 2016. URL: https://books.google.ru/books?id=4Xz-DAAAQBAJ&pg=PT389&lpg=PT389&dq=%D0%BC%D0%B0%D1%81%D1%81%D0%BE%D0%B2%D0%B0%D1%8F+%D0%BA%D0%B0%D1%81%D1%82%D0%BE%D0%BC%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F+%D1%81%D0%BB%D0%BE%D0%B2%D0%B0%D1%80%D1%8C&source=bl&ots=6fTi3A76il&sig=w7Eycx846NVGK_nF2kzev8bKTtU&hl=ru&sa=X&ved=0ahUKEwiGq-Kq49HSAhXHWiwKHXdnDJUQ6AEIPDAG#v=onepage&q&f=false

Электронные ресурсы

1. Konstan J.A. and Ekstrand M.D., (2013). “Recommender systems”. Available at: https://ru.coursera.org/specializations/recommender-systems [Accessed 26 Oct. 2015].
2. Sketch Engine [Электронный ресурс] / Электрон. дан. — Режим доступа: http://www.sketchengine.co.uk/, требуется регистрация.

Приложение 1. Словари

Название словаря	Элементы словаря
цена	цен, деньг, дорого, стоимост, дешев, дешёв, бюджетн, денег
батарея	заряд, заряжают, заряжать, заряжен, заряжая, заряжает, батарее, аккум, блок питан, блоке питан, блоком питан, блока питан, блоки питан, разряжается, разряжался, разряжалась, разряжались, разряжен, разряди, разряжая
видеокарта	видеокарт, дискретн, geforce, фпс, дискретк, видюх, видеоадаптер, видях, видяшк

жёсткий диск	жёсткий диск, жёсткого диск, жёстком диск, жёсткому диск, жесткий диск, жесткого диск, жестком диск, жесткому диск, hdd, ssd, хард, винчестер, нмжд, терабайтный винт, терабайтного винт
внешний вид	дизайн, внешний вид, внешнего вида, смотрится дорого, красивый, стильн, элегантн
габариты	лёгкий, легкий, тонкий, тяжеловат, лёгкост, легкост, вес, габарит, компактност, толщин
звук	звук, колонк, динамик, саб, звучании, громкост
клавиатура	клавиатур, клава, клавы, клавиш, клаве, клаву, клавой
корпус	люфт, скрипи, скрипе, скрипов, скрипа, сборк, маркий, корпус, материал, хлипк, крышк, пластик, трещин, матовый
оперативная память	оперативк, оператива, оперативе, оперативы, оперативой, ddr, оперативная пам, оперативную пам, оперативной пам, озу, ддр, ram
операционная система	win, винда, винде, винду, виндой, винды, операционная сис, операционную сис, операционной сис, ос, виндо, macos, mac os, vista, linux, виста, линукс, убунт, ubuntu, операционк
система охлаждения	вентилятор, кулер, греется, перегревается, охлаждении, термопаст, нагревает
процессор	процессор, железо, железе, железу, железом, железа, ядр, ядер, мощност
скорость работы	производительн, шустр, скорость работы, медленный, тормоз, зависает, подвисает, подтормажив, виснет, висит, быстрота
тачпад	трекпад, трэкпад, тач, touchpad

экран	экран, fullhd, fhd, цветопередач, retina, угол обзор, углы обзор, углами обзор, углах обзор, углов обзор, разрешени, яркост, ips, диспле, дисплэ, ретина, full hd, диагональ, диагонали, монитор

Приложение 2. Фрагменты программы

1) Создание функции анализа отзывов.

Создание списка, в который добавляются отзывы из файла, разбитые на блоки «ссылка», «номер продукта», «оценка», «достоинства», «недостатки», «комментарий»

Создание двух пустых списков, в которые будут записываться оценки для параметров по а) отзывам, б) продуктам.

```
def semAn(f1, f2):
    file1 = open(f1, 'r', encoding='utf-8')
    file2 = open(f2, 'w', encoding='utf-8')
    file1_contents = file1.read()
```

```
list1 = file1_contents.split('\n')
list2 = []
for i in (range(len(list1) - 1)):
    a = list1[i].split('\t')
    list2.append(a)
    b = a[0].split('product/')
    c = b[1].split('/reviews')
    number_of_item = c[0]
    a.insert(1, number_of_item)
list2.sort(key=lambda x: x[1])
k = 1
for i in range(len(list2) - 1):
    if list2[i][1] != list2[i + 1][1]:
        k += 1
list3 = [[0] * 49 for i in range(k)]
comments=0
for i in range(len(list2)):
    comments+=1
list4 = [[0] * 33 for i in range(comments)]
price = open("цена", 'r', encoding='utf-8')
look = open("внешний вид", 'r', encoding='utf-8')
sound = open("звук", 'r', encoding='utf-8')
screen = open("экран", 'r', encoding='utf-8')
battery = open("батарея", 'r', encoding='utf-8')
touch = open("тач", 'r', encoding='utf-8')
speed = open("скорость", 'r', encoding='utf-8')
keyboard = open("клавиатура", 'r', encoding='utf-8')
cooling = open("охлаждение", 'r', encoding='utf-8')
video = open("видеокарта", 'r', encoding='utf-8')
ram = open("оперативка", 'r', encoding='utf-8')
size = open("габариты", 'r', encoding='utf-8')
soft = open("ОС", 'r', encoding='utf-8')
corpus = open("корпус", 'r', encoding='utf-8')
hardware = open("процессор", 'r', encoding='utf-8')
hd = open("винчестер", 'r', encoding='utf-8')
```

2) Сравнение каждой строки словаря (переменная price – отсылает к словарю параметра «Цена») с содержанием блоков «Достоинства» и «Недостатки» для каждого отзыва.

Если соответствие найдено в блоке «Достоинства», в ячейку, которой соответствует параметр «Цена-Достоинство» прибавляется 1. Если соответствие найдено в блоке «Недостатки», 1 добавляется в ячейку, которая отсылает к параметру «Цена-Недостаток». Также увеличивается значение в ячейке, в которой ведётся подсчёт общего количества совпадений, соответствующих данному продукту.

```
for p in price.readlines():
    for i in range(len(list2)):
        for i3 in range(len(list3)):
            p1 = p[:-1]
            if list3[i3][0] == list2[i][1]:
                if list2[i][3].find(str(p1)) != -1:
                    list4[i][1] += 1
                    list3[i3][33] +=1
                if list2[i][4].find(str(p1)) != -1:
                    list4[i][17] +=1
                    list3[i3][33] +=1
```

3) Производится анализ каждой ячейки из списка list3. Содержание ячейки соответствует номеру продукта и количеству положительных и отрицательных оценок для каждого его параметра. То есть 16 ячеек, в которых ведётся подсчёт положительных оценок и 16 ячеек, в которых ведётся подсчёт отрицательных оценок.

Затем высчитывается процент положительных оценок по формуле:

$$\frac{кол\text{-}во\ полож.\ оценок}{кол\text{-}во\ полож.оценок + кол\text{-}во\ отр.оценок} * 100$$

Если процент положительных оценок находится в диапазоне от 0% до 20%, анализируемому параметру присваивается оценка 1, от 20% до 40% - оценка 2, от 40% до 60% - оценка 3, от 60% до 80% - оценка 4, от 80% до 100% - оценка 5.

В том случае, если для выставления оценки по данному параметру недостаточно данных, то есть и количество положительных оценок и количество отрицательных оценок равно нулю, параметру присваивается значение «нет отзывов».

```
for i in list3:
    if int(i[1])!= 0:
        a=int((int(i[1])/(int(i[1])+int(i[17]))*100))
        if a >=80 and a <=100:
            a=5
        elif a>=60 and a<80:
            a=4
        elif a>=40 and a<60:
            a=3
        elif a>=20 and a<40:
            a=2
        else:
            a=1
```

```
    elif int(i[1]) == 0 and int(i[17]) != 0:
        a=1
    else:
        a="нет отзывов"
```

4) Все результаты с пояснениями записываются в файл.

В конце программа закрывает все файлы.

```
        file2.write("Номер продукта: " + str(i[0]) + "; " + "Цена:
" + str(a) + "; " + "Внешний вид: " + str(b) + "; " + "Звук: " +
str(c) + "; " + "Экран: " + str(d) + "; " + "Батарея: " + str(e)
+ "; " + "Тачпад: " + str(f) + "; " + "Скорость и
производительность: " + str(g) + "; " + "Клавиатура: " + str(h) +
"; " + "Система охлаждения: " + str(j) + "; " + "Видеокарта: " +
str(a1) + "; " + "Оперативная память: " + str(b1) + "; " +
"Габариты: " + str(c1) + "; " + "Операционная система: " +
str(d1) + "; " + "Корпус: " + str(e1) + "; " + "Процессор: " +
str(f1) + "; " + "Жёсткий диск: " + str(g1) + '' + '\n')
file1.close()
file2.close()
price.close()
…
```