|  |  |
| --- | --- |
| Подпись и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подпись и дата |  |
| Инв. № подл. | RU. 17701729. 503200-01 81 1-1ЛУ |

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

|  |  |  |
| --- | --- | --- |
| СОГЛАСОВАНО  Доцент факультета компьютерных наук базовой кафедры «Системное программирование» НИУ ВШЭ, канд. физ.-мат. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Д.Ю. Турдаков  «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2017 г. | УТВЕРЖДАЮ   |  | | --- | | Академический руководитель  Образовательной программы  «Программная инженерия» |   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.В.Шилов  «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2017 г. |

**Построение иерархии аспектов по пользовательским отзывам об электронных устройствах**

**Пояснительная записка**

**ЛИСТ УТВЕРЖДЕНИЯ**

RU. 17701729. 503200-0181 1-1ЛУ

Исполнитель: студентка группы БПИ143

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/А.А.Репина/

«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2017 г.

2017

|  |  |
| --- | --- |
| Подпись и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подпись и дата |  |
| Инв. № подл. | RU. 17701729. 503200-0181 |

**УТВЕРЖДЕНО**

**RU. 17701729. 503200-0181 1-1-ЛУ**

|  |  |
| --- | --- |
|  |  |

**Построение иерархии аспектов по пользовательским отзывам об электронных устройствах**

**Пояснительная записка**

RU. 17701729. 503200-0181 1-1

Листов 33

2017

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc477632033)

[НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ 5](#_Toc477632034)

[ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ 6](#_Toc477632035)

[3.1. Постановка цели на разработку 6](#_Toc477632036)

[3.2. Постановка задач на разработку 6](#_Toc477632037)

[3.3. Описание функционирования программы и применяемых алгоритмов 6](#_Toc477632038)

[3.3.1. Описание парсинга входных данных 6](#_Toc477632039)

[3.3.2. Описание используемых для вычисления семантического расстояния характеристик 6](#_Toc477632040)

[3.3.2.1. PMI 6](#_Toc477632041)

[3.3.2.2. Contextual 6](#_Toc477632042)

[3.3.2.3. Syntactic 7](#_Toc477632043)

[3.3.2.4. Lexical 7](#_Toc477632044)

[3.3.3. Описание вычисления семантического расстояния 7](#_Toc477632045)

[3.3.4. Описание построения иерархии 7](#_Toc477632046)

[3.4. Описание и обоснование выбора состава программных средств 8](#_Toc477632047)

[ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ 9](#_Toc477632048)

[ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ 10](#_Toc477632049)

[ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ 11](#_Toc477632050)

[ПРИЛОЖЕНИЕ 1 12](#_Toc477632051)

[ПРИЛОЖЕНИЕ 2 13](#_Toc477632052)

# ВВЕДЕНИЕ

* 1. **Название программы**

Наименование: «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах»

* 1. **Основание для разработки**

Приказ НИУ ВШЭ № 2.3-02/0812-01 от 08.12.2016. Тема работы: «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах». Национальный Исследовательский университет «Высшая школа экономики», факультет компьютерных наук, отделение программной инженерии.

# НАЗНАЧЕНИЕ И ОБЛАСТЬ ПРИМЕНЕНИЯ

1. **Назначение программы**

Данная работа представляет собой подход организации различных аспектов продукта, относящегося к категории электронные устройства, в иерархию на основе знаний о потребительских отзывах. Основываясь на производной иерархии (построенной вручную), создается иерархическая организация опросов потребителей по различным аспектам продукта и совокупным мнениям потребителей по этим аспектам. При такой организации пользователь может получить обзор потребительских мнений в максимально короткий срок.

**2.2. Краткая характеристика области применения программы**

С быстро растущей электронной торговлей большинство розничных веб-сайтов побуждают потребителей писать обзоры, чтобы выразить свои взгляды на различные аспекты продуктов. Огромная коллекция отзывов потребителей теперь доступна в Интернете. Эти обзоры стали важным ресурсом для потребителей и бизнеса. Потребители обычно ищут информацию о качестве в онлайн-опросах потребителей перед покупкой продукта, в то время как многие компании используют онлайн опросы в качестве важного ресурса в их разработке продукта, маркетинге и управлении взаимоотношениями с клиентами. Однако обзоры дезорганизованы, что приводит к трудностям в навигации информации и приобретении знаний. Пользователю нецелесообразно изучать обзоры потребительских мнений по различным аспектам продукта из тысяч источников. Среди аспектов продукта также неэффективно для пользователя просматривать данные и мнения потребителей по определенному аспекту. Таким образом, существует острая необходимость в организации опросов потребителей, чтобы превратить обзоры в полезную структуру знаний. Поскольку иерархия может улучшить представление и доступность информации, то видится разумным организовать аспекты продукта в иерархии и, соответственно, создать иерархическую организацию опросов клиентов.

Чтобы автоматически получить иерархию аспектов из обзоров, можно было бы обратиться к традиционным методам генерации иерархии, которые сначала идентифицируют понятия из текста, а затем определяют отношения между родителем и ребенком. Тем не менее, основанные на шаблонах методы обычно страдают от несогласованности отношений «родитель-потомок» между понятиями, в то время как методы на основе кластеризации часто приводят к низкой точности. Таким образом, путем непосредственного использования этих методов для создания иерархии аспектов из обзоров потребителей полученная иерархия обычно неточна, что приводит к неудовлетворительной организации обзора.

Данная программа предназначена для построения иерархии аспектов по пользовательским отзывам об электронных устройствах на основе производной иерархии, построенной вручную, что пользоволяет максимально быстро предоставить пользователю структурированные данные потребительских мнений.

# ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

## 3.1. Постановка цели на разработку

Ставилась цель создания программы для построения иерархии аспектов по пользовательским отзывам об электронных устройствах.

## 3.2. Постановка задач на разработку

Для достижения данной цели были поставлены следующие задачи:

* 1. Выбрать средства разработки;
  2. Изучить особенности выбранного средства разработки - PyCharm на языке Python;
  3. Разработать архитектуру приложения;
  4. Получить входные данные путем загрузки пользовательских отзывов из Интернета;
  5. Создать требуемые алгоритмы;
  6. Непосредственно создать программу;
  7. Протестировать итоговый продукт.

## 3.3. Описание функционирования программы и применяемых алгоритмов

3.3.1. Описание парсинга входных данных

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

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

Для работы был выбран сайт ulmart.ru, а именно раздел электронных товаров. Для получения отзывов со всех страниц товаров данного сайта был написан парсер, выявляющие отзывы и добавляющий их в базу данных. Всего было получено 24093 отзыва. Отзыв представляет собой объединение 3 частей: плюсы, минусы и комментарий, однако в работе данное разделение учтено не было. Также была создана база данных с 415 идеальными аспектами – теми, что указываются, как описание характеристик того или иного товара. Все вышеуказанные данные и являются входными в созданной программе.

3.3.2. Описание использование ИСП РАН API

Для решения задач, связанных с анализом текстовых данных и их предобработкой, было принято решение об использовании API, предоставляемого ИСП РАН. Благодаря API решались следующие задачи: построение синтаксического дерева предложения, деление текста на предложения и выявление частей речи. Для построения синтаксического дерева в метод, вызывающий API, передается предложение, а на выходе получается строка в формате json. Для деления текста отзыва на отдельные предложения вызывается соответствующий метод и в итоге получаются требуемые данные. Выявление частей речи в передаваемом тексте также осуществляется усилиями API. Подробная документация представлена на сайте: https://api.ispras.ru/texterra/v3.1/docs.

3.3.3. Описание выявления аспектов

Процессу выявления аспектов предшествовал пункт 3.3.1. В качестве входных данных имеется 415 идеальных аспектов и набор отзывов в количестве 24093. Так как отзывы, полученные с ulmart.ru, приходят зашумленными и содержат в себе большое количество случайных символов, то первым делом производится очистка входных данных. Далее производится выявление частей речи всех слов в каждом из отзывов. Данная обработка требуется для того, чтобы в дальнейшем осуществить поиск слова-существительного или набора слов, где главным словом также является существительное. Во втором случае также используется метод построения синтаксического дерева предложения, что позволяет найти нужные связи.

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

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

В дальнейшем для ускорения разработки программы количество аспектов было еще более сужено до 1000 штук, которыми являются самые частые слова и словосочетания. Последующие результаты и числа будут актуальны в рамках данной 1000.

3.3.4. Описание используемых для вычисления семантического расстояния характеристик

3.3.4.1. PMI

PMI включает в себя 2 характеристики: для отзывов и для предложений.

Для вычисления обоих производится итерация по всем парам аспектов, далее производится поиск количества предложений, где есть оба аспекта из пары. Значение PMI считается как логарифм от количества предложений, где есть оба аспекта из пары, деленный на произведение их предвычисленного количества по-отдельности.

Разница в характеристиках заключается в корпусе отзывов/предложений, который передается для дальнейшей обработки в метод расчёта PMI.

3.3.4.2. Contextual

Данная характеристика состоит из двух значений: локального и глобального.

Для получения локального контекста для каждой пары аспектов были предварительно вычислены значения 4 окружающих каждый из аспектов слов в отзывах: двух справа и двух слева. Из данных предварительных значений была сформирована база данных, которая позволила повысить скорость дальнейших вычислений.

Для вычисления глобального контекста для каждого из аспектов в паре был сформирован корпус отзывов, где он встречался. Далее, как для глобального, так и для локального контекста были загружены предвычисленные данные из баз данных в память, для получения результатов, поддающихся анализу, был произведен smoothing данных методом Add-1 Smoothing[[1]](#footnote-1).

После было осуществлено вычисления значения, именуемого kl divergence для локального и глобального контекстов, которое означает, насколько контексты аспектов в паре удалены друг от друга.

3.3.4.3. Syntactic

Для вычисления данной величины были предварительно сохранены в базу данных синтаксические деревья, полученные с помощью API ИСП РАН[[2]](#footnote-2), для всех предложений из отзывов. Далее для каждой из пар аспектов были найдены предложения, где встречаются они оба. Во всех таких предложениях с помощью вышеуказанной базы данных был вычислен кратчайший путь от одного аспекта к другому. Далее для каждой из пар был вычислен средний такой путь через деление суммы путей на их количество.

3.3.4.4. Lexical

Для вычисления данной величины были проитерированы все пары аспектов и вычислена разница между их длинами.

3.3.5. Описание вычисления семантического расстояния

Для вычисления семантической дистанции было выбрано два способа расчётов, один из которых был приведен в статье, которая легла в основу данной работы, а второй является примером использования методов машинного обучения на практике.

В первом способе производится вычисление вектора w, размера 1Х6, представляющего собой набор характеристик (PMI, Lexical, Syntactic, Contextual) идеальных аспектов. Данный вектор вычисляется по формуле:

,

где f – вектор характеристик, η – константное значение, равное 0.4, I – единичная матрица, d – вектор, содержащий в себе все кратчайшие пути между идеальными аспектами в построенном вручную дереве. Далее для каждой пары аспектов семантическое расстояние равно:

,

где w - вектор, вычисленный на шаг раньше, а другие множители являются значениями характеристик для конкретной пары аспектов, которые получаются путем извлечения из баз данных.

Второй способ представляет собой пример использования RandomForestRegressor, который являет частью пакета библиотеки sklearn.ensemble. В качестве входных данных для обучения модели поступают результаты идеальных аспектов, а именно их предвычисленные реальные длины путей для каждой пары аспектов, а также набор из вышеупомянутых 6 характеристик для каждой пары аспектов. Обучившись модель получает задание предсказать результаты уже для реальных данных.

Проведенное изучение двух подходов к вычислению семантического расстояния показало наличие, как преимуществ, так и недостатков у обоих. Для способа 1 данные расстояний угадываются верно существенно реже, нежели чем в способе 2, однако способ 2 сильно зависит от тренировочной выборки и поэтому не имеет возможности выдавать результаты, выходящие за пределы предложенных ему для обучения, что является существенным минусом, коим не страдает 1 способ. Также за счет большего диапазона значений 1 способ является более удобным в плане построения дальнейшей иерархии аспектов, однако для полноты картины в пункте 3.3.4. были использованы и результаты 2 подхода для сравнения. В качестве преимущества второго способа можно указать скорость его работы, которая в разы выше нежели чем у 1 подхода.

3.3.6. Описание построения иерархии

Для построения иерархии аспектов было вручную построено дерево сначала для идеальных аспектов. Далее алгоритм для каждого из свободных листов дерева(родителей) выполняет поиск подходящих продолжений(детей) на основе того, насколько семантическая дистанция отличается от среднего значения дистанции для изначального дерева идеальных аспектов. Каждый новый ребенок запоминается в базу данных вместе со своим родителем и добавляется в список потенциально возможных родителей для новых аспектов.

## 3.4. Описание и обоснование выбора состава программных средств

Программа предназначена для операционной системы с установленным Python 2.7 или Python 3.2+., так как написана на языке программирования Python.

# ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

1. **Ориентировочная экономическая эффективность**

Программа помогает упростить процесс организации опросов потребителей, чтобы превратить обзоры в полезную структуру знаний. Поскольку иерархия может улучшить представление и доступность информации, то видится разумным организовать аспекты продукта в иерархии и, соответственно, создать иерархическую организацию опросов клиентов. При такой организации пользователь может получить обзор потребительских мнений в максимально короткий срок.

1. **Экономические преимущества разработки по сравнению с аналогами**

Имеется несколько альтернативных программ, работающих с англоязычным сегментом интернета, например, HASM (Hierarchical Aspect-Sentiment Model), JST (Joint modeling of Sentiment and Topic), ASUM (Aspect Sentiment Uniﬁcation Model), JST (Joint Sentiment-Topic). Однако похожей программы для обработки русскоязычных пользовательских отзывов и представления их в виде иерархии аспектов нет.

# ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. Сайт материалов университета Иллинойс [Электронный ресурс]. URL: https://courses.engr.illinois.edu/cs498jh/Slides/Lecture03.pdf (дата обращения: 20.12.2016);
2. Сайт API Института системного программирования Российской академии наук [Электронный ресурс]. URL: https://api.ispras.ru/ (дата обращения: 22.11.2016);
3. С. Ким, Д. Цанг, Ж. Чен, Э. Оу, Ш. Лиу, “A Hierarchical Aspect-Sentiment Model for Online Reviews”, Департамент компьютерных наук, Корея, 2010;
4. Дж. Ю, Ж. Джа, М. Венг, К. Венг, Т. Чуа, “Domain-Assisted Product Aspect Hierarchy Generation: Towards Hierarchical Organization of Unstructured Consumer Reviews”, 2011;
5. В. Проноза, Е.В. Ягунова, “Аспектный анализ отзывов о ресторанах для рекомендательных систем е-туризма”, Санкт-Петербургский государственный университет, Санкт-Петербург, 2010;
6. А.А. Бреслав, А.П. Лукьянова, М.А. Коротков, “Построение иерархии классов по текстовым описаниям“, Санкт-Петербургский государственный политехнический университет, Санкт-Петербург, 2011;
7. Сайт документации языка Python [Электронный ресурс]. URL: https://docs.python.org/3/ (дата обращения: 20.01.2017).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ | | | | | | | | | | |
| Изм. |  | Номера листов (страниц) | | | | Всего  листов  (страниц) в  документе | №  документа | Входящий №  сопроводительн  ого документа и  дата | Подпись | Дата |
| измененн  ых | | замененны  х | новых | аннулиро  ванных |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |

# ПРИЛОЖЕНИЕ 1

**Описание и функциональное назначение классов и интерфейсов**

|  |  |
| --- | --- |
| Класс или интерфейс | Назначение |
| Aspects | Отвечает за поиск аспектов в отзывах пользователей и морфологический разбор предложений |
| Context | Отвечает за расчет характеристик Context Global и Context Local для пар аспектов |
| DB | Отвечает за создание и взаимодействие со всеми базами данных |
| FrequentAspects | Отвечает за поиск 1000 наиболее популярных аспектов |
| HierarchyBuilder | Отвечает за построение иерархии аспектов по пользовательским отзывам о электронных устройствах |
| IdealAspectsDB | Отвечает за заполнение базы данных с идеальными аспектами |
| Lexical | Отвечает за расчет характеристики Lexical для пар аспектов |
| Main | Является точкой управления программой. Отвечает за запуск работы всех классов |
| OneClassSVM | Отвечает за One Class SVM тренировку и обучение |
| PMI | Отвечает за расчет характеристик PMI Review и PMI Sentence для пар аспектов |
| SemanticDistanceLearning | Отвечает за вычисление семантического расстояния |
| RandomForest | Отвечает за вычисление семантического расстояния методами машинного обучения |
| Sentence | Отвечает за разделение пользовательских отзывов на предложения |
| Splitter | Отвечает за очистку отзывов и предложений от мусора |
| Syntactic | Отвечает за расчет характеристики Syntactic для пар аспектов |
| Unnecessary | Отвечает за удаление ненужных частей предложений |
| CategoryNames | Отвечает за создание списка категорий сайта ulmart.ru |
| DataBase\_Ulmart | Отвечает за формирование базы отзывов |
| Parser | Класс-парсер сайта ulmart.ru |

# ПРИЛОЖЕНИЕ 2

**Описание и функциональное назначение методов, полей и свойств, вложенных типов**

1. Aspects.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| api\_key | protected | string | API ключ |
| url\_syntatic\_parsing | protected | string | Ссылка для синтаксического парсинга |
| url\_pos | protected | string | Ссылка для определения части речи |
| url\_sentence | protected | string | Ссылка для разделения текста на предложения |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process | protected | aspect, db | void | Запуск работы всех методов |
| syntatic\_parsing | protected | review, aspect | string | Построение синтаксического дерева |
| aspects\_find | protected | part, aspect | list | Поиск аспектов |
| one\_word | protected | data, item, list\_aspects, pos\_arr | list | Аспект, состоящий из 1 слова |
| word\_pair | protected | data, item, list\_aspects, pos\_arr | list | Аспект, состоящий из нескольких слов |
| replacer | protected | item | string | Очистка строки от ненужных символов |
| parse\_pos | protected | pos\_sentence | string | Определение части речи |
| tag\_part\_of\_speech | protected | item, aspect | void | Запрос к API для определения части речи |
| move\_ideal\_aspects | protected | ideal, ideal\_aspects, db | void | Перемещение аспектов |
| get\_ideal | protected | part, ideal\_aspects | list | Получение списка идеальных аспектов |

1. Context.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process | protected | db, aspects | void | Запуск методов класса для обычных аспектов |
| process\_ideal | protected | db | void | Запуск методов класса для идеальных аспектов |
| get\_ideal\_dict | protected |  | ditctionary | Получение словаря идеальных аспектов |
| get\_reviews\_and\_vocabulary | protected | db, all\_aspects\_words | list | Получение отзывов и обычных аспектов |
| form\_local\_context\_db | protected | db, aspects, reviews, which\_part | void | Создание базы данных локального контекста обычных аспектов |
| form\_global\_context\_db | protected | db, aspects, reviews, which\_part | void | Создание базы данных глобального контекста обычных аспектов |
| form\_global\_context\_extra\_db | protected | db, aspects | void | Создание  дополнительной базы данных глобального контекста обычных аспектов |
| form\_global\_context\_extra\_ideal\_db | protected | db, aspects | void | Создание  дополнительной базы данных глобального контекста идеальных аспектов |
| is\_several\_word\_aspect\_in\_review | protected | clear\_aspect\_words, review, str\_context, is\_global | string | Проверка наличия нескольких слов аспекта в отзыве |
| is\_one\_word\_aspect\_in\_review | protected | aspect, review, str\_context, is\_global | string | Проверка наличия однословного аспекта в отзыве |
| form\_str\_context | protected | index, words, str\_context | string | Создание строки контекста |
| check\_left\_index | protected | index, words | string | Проверка левого индекса |
| check\_right\_index | protected | index, words | string | Проверка правого индекса |
| replacer | protected | item | string | Очистка данных |
| local\_context | protected | db, all\_aspects\_words | void | Вычисление локального контекста обычных аспектов |
| local\_context\_ideal | protected | db, all\_aspects\_words | void | Вычисление локального контекста идеальных аспектов |
| global\_context | protected | db, all\_aspects\_words | void | Вычисление глобального контекста обычных аспектов |
| global\_context\_ideal | protected | db, all\_aspects\_words | void | Вычисление глобального контекста идеальных аспектов |
| add\_one\_smoothing | protected | list | list | Smoothing данных |

1. DB.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| conn\_aspects | protected | Connection | Соединение с базой данных |
| conn\_reviews | protected | Connection | Соединение с базой данных |
| conn\_merged | protected | Connection | Соединение с базой данных |
| conn\_sentence | protected | Connection | Соединение с базой данных |
| conn\_aspects\_one\_word | protected | Connection | Соединение с базой данных |
| conn\_reviews\_one\_word | protected | Connection | Соединение с базой данных |
| conn\_sentences\_one\_word | protected | Connection | Соединение с базой данных |
| conn\_pmi\_review | protected | Connection | Соединение с базой данных |
| conn\_pmi\_sentence | protected | Connection | Соединение с базой данных |
| conn\_pmi\_ideal\_review | protected | Connection | Соединение с базой данных |
| conn\_pmi\_ideal\_sentence | protected | Connection | Соединение с базой данных |
| conn\_path\_weight | protected | Connection | Соединение с базой данных |
| conn\_semantic\_distance | protected | Connection | Соединение с базой данных |
| conn\_semantic\_distance\_ideal | protected | Connection | Соединение с базой данных |
| conn\_local\_context\_prepare | protected | Connection | Соединение с базой данных |
| conn\_local\_context\_prepare\_ideal | protected | Connection | Соединение с базой данных |
| conn\_global\_context\_prepare | protected | Connection | Соединение с базой данных |
| conn\_global\_context\_prepare\_extra | protected | Connection | Соединение с базой данных |
| conn\_global\_context\_prepare\_ideal | protected | Connection | Соединение с базой данных |
| conn\_global\_context\_prepare\_extra\_ideal | protected | Connection | Соединение с базой данных |
| conn\_global\_context | protected | Connection | Соединение с базой данных |
| conn\_local\_context | protected | Connection | Соединение с базой данных |
| conn\_lexical | protected | Connection | Соединение с базой данных |
| conn\_lexical\_ideal | protected | Connection | Соединение с базой данных |
| conn\_syntactic | protected | Connection | Соединение с базой данных |
| conn\_syntactic\_ideal | protected | Connection | Соединение с базой данных |
| conn\_tree | protected | Connection | Соединение с базой данных |
| conn\_local\_context\_ideal | protected | Connection | Соединение с базой данных |
| conn\_global\_context\_ideal | protected | Connection | Соединение с базой данных |
| conn\_hierarchy | protected | Connection | Соединение с базой данных |
| conn\_frequent | protected | Connection | Соединение с базой данных |
| conn\_ideal\_full | protected | Connection | Соединение с базой данных |
| conn\_hierarchy\_real | protected | Connection | Соединение с базой данных |
| conn\_semantic\_distance\_real | protected | Connection | Соединение с базой данных |
| cursor\_aspects | protected | Cursor | Курсор базы данных |
| cursor\_aspects2 | protected | Cursor | Курсор базы данных |
| cursor\_reviews | protected | Cursor | Курсор базы данных |
| cursor\_article | protected | Cursor | Курсор базы данных |
| cursor\_merged | protected | Cursor | Курсор базы данных |
| cursor\_sentence | protected | Cursor | Курсор базы данных |
| cursor\_aspects\_one\_word | protected | Cursor | Курсор базы данных |
| cursor\_reviews\_one\_word | protected | Cursor | Курсор базы данных |
| cursor\_reviews\_one\_word\_update | protected | Cursor | Курсор базы данных |
| cursor\_sentences\_one\_word | protected | Cursor | Курсор базы данных |
| cursor\_sentences\_one\_word\_update | protected | Cursor | Курсор базы данных |
| cursor\_pmi\_review | protected | Cursor | Курсор базы данных |
| cursor\_pmi\_sentence | protected | Cursor | Курсор базы данных |
| cursor\_pmi\_ideal\_review | protected | Cursor | Курсор базы данных |
| cursor\_pmi\_ideal\_sentence | protected | Cursor | Курсор базы данных |
| cursor\_path\_weight | protected | Cursor | Курсор базы данных |
| cursor\_semantic\_distance | protected | Cursor | Курсор базы данных |
| cursor\_semantic\_distance\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_local\_context\_prepare | protected | Cursor | Курсор базы данных |
| cursor\_local\_context\_prepare\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_global\_context\_prepare | protected | Cursor | Курсор базы данных |
| cursor\_global\_context\_prepare\_extra | protected | Cursor | Курсор базы данных |
| cursor\_global\_context\_prepare\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_global\_context\_prepare\_extra\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_global\_context | protected | Cursor | Курсор базы данных |
| cursor\_local\_context\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_global\_context\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_local\_context | protected | Cursor | Курсор базы данных |
| cursor\_lexical | protected | Cursor | Курсор базы данных |
| cursor\_lexical\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_syntactic | protected | Cursor | Курсор базы данных |
| cursor\_syntactic\_ideal | protected | Cursor | Курсор базы данных |
| cursor\_tree | protected | Cursor | Курсор базы данных |
| cursor\_hierarchy | protected | Cursor | Курсор базы данных |
| cursor\_frequent | protected | Cursor | Курсор базы данных |
| cursor\_ideal\_full | protected | Cursor | Курсор базы данных |
| cursor\_hierarchy\_real | protected | Cursor | Курсор базы данных |
| cursor\_semantic\_distance\_real | protected | Cursor | Курсор базы данных |
| db\_merged\_name | protected | string | Имя базы данных |
| db\_aspects\_name | protected | string | Имя базы данных |
| db\_reviews\_name | protected | string | Имя базы данных |
| db\_sentence\_name | protected | string | Имя базы данных |
| db\_aspects\_one\_word\_name | protected | string | Имя базы данных |
| db\_reviews\_one\_word\_name | protected | string | Имя базы данных |
| db\_sentences\_one\_word\_name | protected | string | Имя базы данных |
| db\_pmi\_review\_name | protected | string | Имя базы данных |
| db\_pmi\_sentence\_name | protected | string | Имя базы данных |
| db\_pmi\_ideal\_review\_name | protected | string | Имя базы данных |
| db\_pmi\_ideal\_sentence\_name | protected | string | Имя базы данных |
| db\_path\_weight | protected | string | Имя базы данных |
| db\_semantic\_distance | protected | string | Имя базы данных |
| db\_semantic\_distance\_ideal | protected | string | Имя базы данных |
| db\_local\_context\_prepare | protected | string | Имя базы данных |
| db\_local\_context\_prepare\_ideal | protected | string | Имя базы данных |
| db\_global\_context\_prepare | protected | string | Имя базы данных |
| db\_global\_context\_prepare\_extra | protected | string | Имя базы данных |
| db\_global\_context\_prepare\_ideal | protected | string | Имя базы данных |
| db\_global\_context\_prepare\_extra\_ideal | protected | string | Имя базы данных |
| db\_global\_context | protected | string | Имя базы данных |
| db\_local\_context | protected | string | Имя базы данных |
| db\_lexical | protected | string | Имя базы данных |
| db\_lexical\_ideal | protected | string | Имя базы данных |
| db\_syntactic | protected | string | Имя базы данных |
| db\_syntactic\_ideal | protected | string | Имя базы данных |
| db\_tree | protected | string | Имя базы данных |
| db\_local\_context\_ideal | protected | string | Имя базы данных |
| db\_global\_context\_ideal | protected | string | Имя базы данных |
| db\_hierarchy | protected | string | Имя базы данных |
| db\_frequent | protected | string | Имя базы данных |
| db\_ideal\_full | protected | string | Имя базы данных |
| db\_hierarchy\_real | protected | string | Имя базы данных |
| db\_semantic\_distance\_real | protected | string | Имя базы данных |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| create\_frequent\_db | protected |  | void | Создание базы данных |
| create\_hierarchy\_db | protected |  | void | Создание базы данных |
| create\_context\_global\_ideal\_db | protected |  | void | Создание базы данных |
| create\_context\_local\_ideal\_db | protected |  | void | Создание базы данных |
| create\_tree\_db | protected |  | void | Создание базы данных |
| create\_syntactic\_ideal\_db | protected |  | void | Создание базы данных |
| create\_syntactic\_db | protected |  | void | Создание базы данных |
| create\_lexical\_db | protected |  | void | Создание базы данных |
| create\_lexical\_ideal\_db | protected |  | void | Создание базы данных |
| create\_context\_global\_db | protected |  | void | Создание базы данных |
| create\_context\_local\_db | protected |  | void | Создание базы данных |
| create\_context\_local\_prepare\_db | protected |  | void | Создание базы данных |
| create\_context\_local\_prepare\_ideal\_db | protected |  | void | Создание базы данных |
| create\_context\_global\_prepare\_ideal\_db | protected |  | void | Создание базы данных |
| create\_context\_global\_prepare\_extra\_ideal\_db | protected |  | void | Создание базы данных |
| create\_context\_global\_prepare\_db | protected |  | void | Создание базы данных |
| create\_context\_global\_prepare\_extra\_db | protected |  | void | Создание базы данных |
| create\_semantic\_distance\_db | protected |  | void | Создание базы данных |
| create\_semantic\_distance\_ideal\_db | protected |  | void | Создание базы данных |
| create\_path\_weight\_db | protected |  | void | Создание базы данных |
| create\_pmi\_ideal\_review\_db | protected |  | void | Создание базы данных |
| create\_pmi\_ideal\_sentence\_db | protected |  | void | Создание базы данных |
| create\_pmi\_review\_db | protected |  | void | Создание базы данных |
| create\_pmi\_sentence\_db | protected |  | void | Создание базы данных |
| create\_aspects\_one\_word\_db | protected |  | void | Создание базы данных |
| create\_reviews\_one\_word\_db | protected |  | void | Создание базы данных |
| create\_sentences\_one\_word\_db | protected |  | void | Создание базы данных |
| create\_aspects\_db | protected |  | void | Создание базы данных |
| create\_sentence\_db | protected |  | void | Создание базы данных |
| create\_semantic\_distance\_real\_db | protected |  | void | Создание базы данных |
| create\_hierarchy\_real\_db | protected |  | void | Создание базы данных |
| create\_ideal\_full\_db | protected |  | void | Создание базы данных |
| add\_semantic\_distance\_real | protected | aspect1, aspect2, distance | void | Добавление данных в базу данных |
| add\_hierarchy\_real | protected | parent, child | void | Добавление данных в базу данных |
| add\_ideal\_full | protected | filename, aspect1, aspect2, pmi\_review, pmi\_sentence, lexical, syntactic, local\_context, global\_context, weight | void | Добавление данных в базу данных |
| add\_frequent | protected | word, number | void | Добавление данных в базу данных |
| add\_hierarchy | protected | parent, child | void | Добавление данных в базу данных |
| add\_tree | protected | sentence, tree | void | Добавление данных в базу данных |
| add\_syntactic\_ideal | protected | aspect1, aspect2, syntactic\_path | void | Добавление данных в базу данных |
| add\_syntactic | protected | aspect1, aspect2, syntactic\_path | void | Добавление данных в базу данных |
| add\_lexical | protected | aspect1, aspect2, length\_difference | void | Добавление данных в базу данных |
| add\_lexical\_ideal | protected | aspect1, aspect2, length\_difference | void | Добавление данных в базу данных |
| add\_context\_global\_ideal | protected | aspect1, aspect2, kl\_divergence | void | Добавление данных в базу данных |
| add\_context\_local\_ideal | protected | aspect1, aspect2, kl\_divergence | void | Добавление данных в базу данных |
| add\_context\_global | protected | aspect1, aspect2, kl\_divergence | void | Добавление данных в базу данных |
| add\_context\_local | protected | aspect1, aspect2, kl\_divergence | void | Добавление данных в базу данных |
| add\_context\_local\_prepare | protected | aspect, context | void | Добавление данных в базу данных |
| add\_context\_local\_prepare\_ideal | protected | aspect, context | void | Добавление данных в базу данных |
| add\_context\_global\_prepare\_ideal | protected | aspect, review | void | Добавление данных в базу данных |
| add\_context\_global\_prepare\_extra\_ideal | protected | aspect, context | void | Добавление данных в базу данных |
| add\_context\_global\_prepare | protected | aspect, review | void | Добавление данных в базу данных |
| add\_context\_global\_prepare\_extra | protected | aspect, context | void | Добавление данных в базу данных |
| add\_semantic\_distance\_ideal | protected | aspect1, aspect2, distance | void | Добавление данных в базу данных |
| add\_semantic\_distance | protected | aspect1, aspect2, distance | void | Добавление данных в базу данных |
| add\_path\_weight | protected | filename, aspect1, aspect2, weight | void | Добавление данных в базу данных |
| add\_pmi\_ideal\_review | protected | aspect1, aspect2, num1, num2, both\_num, pmi | void | Добавление данных в базу данных |
| add\_pmi\_ideal\_sentence | protected | aspect1, aspect2, num1, num2, both\_num, pmi | void | Добавление данных в базу данных |
| add\_pmi\_review | protected | aspect1, aspect2, num1, num2, both\_num, pmi | void | Добавление данных в базу данных |
| add\_pmi\_sentence | protected | aspect1, aspect2, num1, num2, both\_num, pmi | void | Добавление данных в базу данных |
| add\_review | protected | article, advantage\_aspects, disadvantage\_aspects, comment\_aspects | void | Добавление данных в базу данных |
| add\_one\_word\_aspects | protected | article, advantage\_aspects, disadvantage\_aspects, comment\_aspects | void | Добавление данных в базу данных |
| add\_one\_word\_review | protected | article, advantage\_aspects, disadvantage\_aspects, comment\_aspects | void | Добавление данных в базу данных |
| add\_one\_word\_sentence | protected | article, sentence | void | Добавление данных в базу данных |
| delete\_aspects | protected | article | void | Удаление записи |

1. FrequentAspects.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process | protected | db | void | Сбор 1000 самых популярных аспектов в базу данных |
| replacer | protected | item | string | Очистка 1000 самых популярных аспектов от мусора |

1. IdealAspectsDB.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| conn\_aspects | protected | Connection | Соединение с базой данных |
| cursor\_aspects | protected | Cursor | Курсор базы данных |
| cursor\_aspects\_update | protected | Cursor | Курсор базы данных для обновления |
| db\_aspects\_name | protected | string | Имя базы данных |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| create\_aspects\_db | protected |  | void | Создание базы данных с аспектами |
| add\_review | protected | article, advantage\_aspects, disadvantage\_aspects, comment\_aspects | void | Добавление отзыва в базу данных |
| remove\_duplicates | protected |  | void | Удаление дублей |
| process | protected | part | string | Сбор частей в единое целое |
| count\_aspects | protected |  | void | Подсчет количества аспектов |

1. HierarchyBuilder.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| find\_free\_nodes | protected |  | list | Поиск свободных листьев |
| build\_hierarchy | protected | db, average\_semantic\_distance\_ideal |  | Построение иерархии |
| process | protected | db, average\_semantic\_distance\_ideal |  | Запуск методов класса |
| calculate\_average\_semantic\_distance\_ideal\_tree | protected | db | float | Вычисление среднего расстояния между идеальными аспектами |

1. Main.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| main | protected | Main | Доступ к методам класса Main |
| vocabulary | protected | dictionary | Список аспектов |
| db | protected | DB | Доступ к методам класса DB |
| aspect | protected | Aspect | Доступ к методам класса Aspect |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| aspects\_process | protected |  | void | Запуск работы класса аспектов |
| sentence\_process | protected |  | void | Запуск работы класса предложений |
| split\_process | protected |  | void | Запуск работы класса очистки |
| pmi\_process | protected |  | void | Запуск работы методов PMI |
| pmi\_review | protected | pmi, reviews\_corpus, vocabulary | void | Запуск работы  PMI review |
| pmi\_sentence | protected | pmi, sentences\_corpus, vocabulary | void | Запуск работы  PMI sentence |
| pmi\_ideal | protected | pmi, reviews\_corpus, sentences\_corpus | void | Запуск работы  PMI ideal |
| semantic\_learning\_process | protected |  | void | Запуск работы класса расчета семантического расстояния |
| contextual\_features | protected |  | void | Запуск работы класса Context |
| lexical\_features | protected |  | void | Запуск работы класса Lexical |
| syntactic\_features | protected |  | void | Запуск работы класса Syntactic |
| frequent | protected |  | void | Запуск работы класса поиска самых частотных аспектов |
| hierarchy | protected |  | void | Запуск работы класса построения иерархии |

1. Lexical.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process | protected | aspects, db | void | Вычисление характеристики Lexical для аспектов |
| process\_ideal | protected | db | void | Вычисление характеристики Lexical для идеального аспектов |

1. OneClassSVM.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| get\_data | protected | db | list | Получение данных (аспектов) |
| get\_labels | protected | data, db | list | Получение меток (Имен категорий) |
| get\_ideal\_data | protected | data, labels | list | Получение идеальных данных (идеальных аспектов) |
| unarray | protected | data | string | Разархивация данных |
| train\_and\_predict | protected | train\_data, test\_data | vector | Тренировать и предсказать |

1. PMI.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| get\_all\_ideal\_aspects\_from\_train\_files | protected |  | list | Получение списка всех идеальных аспектов |
| iterate\_ideal\_aspects\_files | protected | pmi, reviews\_corpus, sentences\_corpus, db | void | Создание PMI базы данных идеальных аспектов |
| get\_all\_reviews\_corpus | protected | db | list | Получение всех отзывов |
| get\_all\_sentences\_corpus | protected | db | list | Получение всех предложений |
| get\_vocabulary | protected | db | dictionary | Получение всех аспектов |
| get\_frequent\_vocabulary | protected | db | dictionary | Получение 1000 самых популярных аспектов |
| one\_word\_aspects | protected | ideal, db | void | Создание однословных аспектов |
| one\_word\_reviews | protected | db | void | Создание однословных отзывов на основе однословных аспектов |
| process\_review | protected | part, aspects | string | Обработка отзывов |
| create\_one\_word\_list | protected | part | list | Создание однословного списка |
| calculate\_pmi | protected | corpus, which\_part, vocabulary, db | void | Вычисление PMI характеристики |
| create\_col\_array | protected | matrix, matrix\_terms\_len | array | Создание двумерного массива из матрицы |

1. SemanticDistanceLearning.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение | |
| calculate\_ground\_truth\_distance | protected | db | void | Расчет минимального пути между парами аспектов в дереве | |
| find\_path | protected | node,  next\_node,  filename | int | Поиск пути между двумя узлами | |
| calculate\_distance | protected | db | vector | Расчет вектора дистанции на основе характеристик | |
| vector\_with\_ground\_truth\_distances | protected | db | vector | Получение вектора с минимальными путями для всех пар идеальных аспектов | |
| process\_semantic\_distance\_learning | protected | db | void | Запуск расчета семантического расстояния для обычных аспектов | |
| process\_semantic\_distance\_learning\_ideal | protected | db | void | Запуск расчета семантического расстояния для идеальных аспектов | |
| print\_data | protected | db | void | Вывод данных на экран | |

1. Sentence.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process | protected | db, aspect | void | Запуск методов класса |
| process\_one\_word | protected | db, aspect | void | Обработка аспектов из 1 слова в предложении |
| process\_review | protected | part, aspects | string | Замена частей отзыва |
| clean\_sentences | protected | sentences\_from\_api | list | Очистка предложения |
| ask\_api | protected | review, aspect | string | Выполнение запроса к API |

1. Splitter.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process\_reviews | protected | db | void | Запуск очистки отзывов |
| process\_sentences | protected | db | void | Запуск очистки предложений |
| clean | protected | part | string | Очистка строки или ее части |

1. Syntactic.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| process | protected | db, vocabulary, aspect\_class\_object | void | Запуск всех методов класса для обычных аспектов |
| process\_ideal | protected | db | void | Запуск всех методов класса для идеальных аспектов |
| calculate\_syntactic | protected | aspect1, aspect2, non\_zero\_sentences, db | float | Расчет характеристики  Syntactic |
| find\_path | protected | aspect1, aspect2, aspect1\_parent, aspect2\_parent, syntax\_relations, aspect1\_parents, aspect2\_parents | void | Поиск кратчайшего пути от одного аспекта к другому |
| get\_parent | protected | start\_index, syntax\_relations | string | Поиск родителя для аспекта |
| find\_path\_for\_sentence | protected | sentence, aspect1, aspect2, syntactic\_paths\_sum, divider, db | float | Поиск кратчайшего пути в нескольких предложениях |
| build\_tree | protected | sentences, aspect\_class\_object, db | void | Построение синтаксического дерева и сохранение в базе данных |

1. Unnecessary.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| ideal | protected | IdealAspectsDB | Объект класса IdealAspectsDB |
| row\_aspect | protected | cursor | Курсор базы данных |
| count | protected | int | Счетчик строк |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| updater | protected | part | string | Обновляет часть отзыва |

1. CategoryNames.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| conn | protected | сonnection | Соединение с базой данных |
| с | protected | сursor | Курсор для базы данных |
| dbName | protected | string | Имя базы данных |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| site\_directories\_print | protected |  | void | Вывод всех директорий сайта ulmart.ru |
| sub\_categories\_print | protected |  | void | Вывод всех подкатегорий сайта ulmart.ru |
| create | protected | name | void | Создание списка категорий сайта ulmart.ru |

1. DataBase\_Ulmart.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| conn | protected | сonnection | Соединение с базой данных |
| с | protected | сursor | Курсор для базы данных |
| dbName | protected | string | Имя базы данных |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| create\_db | protected |  | void | Создание базы данных |
| add\_review | protected | category\_name, subcategory\_name, article, advantage, disadvantage, comment | void | Добавление отзыва в базу данных |
| reviews\_num | protected | article | cursor | Количество отзывов с конкретным номером |
| total | protected |  | cursor | Общее число отзывов |
| remove\_review | protected | article | void | Удаление отзыва |
| commit | protected |  | void | Сохранение данных в базе |

1. Parser.py

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| parse\_product | protected | start\_link, category\_name, subcategory\_name | void | Парсер конкретного продукта |
| parse\_sub\_category | protected | start\_link, category\_name | void | Парсер подкатегорий |
| parse\_categories | protected |  | void | Парсер категорий |
| reviews | protected | article, category\_name, subcategory\_name | boolean | Загрузчик отзывов пользователей на основе данных парсинга |

1. RandomForest.py

|  |  |  |  |
| --- | --- | --- | --- |
| Поля | | | |
| Имя | Мод.доступа | Тип | Назначение |
| model | protected | Model | Модель для машинного обучения |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Методы | | | | |
| Имя | Мод. доступа | Аргументы | Тип | Назначение |
| get\_test | protected | db | list | Получение тестовых данных |
| form\_train | protected | db | void | Формирование тренировочных данных |
| get\_train | protected | db | list | Получение тренировочных данных |
| process | protected | x\_test, x\_train, y\_train | void | Запуск работы обучения и предсказания |

1. https://courses.engr.illinois.edu/cs498jh/Slides/Lecture03.pdf [↑](#footnote-ref-1)
2. https://api.ispras.ru/ [↑](#footnote-ref-2)