|  |  |
| --- | --- |
| Подпись и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подпись и дата |  |
| Инв. № подл. | RU. 17701729. 503200-01 51 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**

**Программа и методика испытаний**

**ЛИСТ УТВЕРЖДЕНИЯ**

RU. 17701729. 503200-0151 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. 503200-0151 |

**УТВЕРЖДЕНО**

**RU. 17701729. 503200-0151 1-1-ЛУ**

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

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

**Программа и методика испытаний**

RU. 17701729. 503200-0151 1-1

Листов 11

2017

СОДЕРЖАНИЕ

[1. Объект испытаний 3](#_Toc440542745)

[2. Цель испытаний 4](#_Toc440542746)

[3. Требования к программе 5](#_Toc440542747)

[4. Требования к программной документации 6](#_Toc440542748)

[5. Средства и порядок испытаний 7](#_Toc440542749)

[6. Методы испытаний 8](#_Toc440542750)

[Лист регистрации изменений 11](#_Toc440542751)

# ОБЪЕКТ ИСПЫТАНИЙ

1.1 Наименование программы

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

1.2 Область 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.

# ЦЕЛЬ ИСПЫТАНИЙ

Целью испытаний является проверка правильности выполнения программой функций, перечисленных в разделе «Требования к программе».

# ТРЕБОВАНИЯ К ПРОГРАММЕ

Программа должна предоставлять возможность построения иерархии аспектов по пользовательским отзывам об электронных устройствах. Для этого она должна предоставлять возможность выполнения перечисленных ниже функций:

1. метод, принимающий в себя адрес сайта и возвращающий набор отзывов со всех страниц, находящихся по адресу;
2. метод, принимающий на входе корпус аспектов, корпус отзывов и корпус их предложений и возвращающий величины PMI для данных корпусов;
3. метод, принимающий на входе корпус аспектов, корпус отзывов и корпус их предложений и возвращающий величины Context для глобального и локального контекстов;
4. метод, принимающий на входе корпус аспектов и возвращающий величину Lexical для каждой пары аспектов;
5. метод, принимающий на входе корпус аспектов, корпус предложений и их синтаксических деревьев и возвращающий величину Syntactic для каждой пары аспектов;
6. метод, принимающий на вход корпус аспектов, корпус характеристик (вычисленных для каждой пары) и возвращающий семантическое расстояние для каждой пары аспектов;
7. метод, принимающий на входе корпус аспектов и корпус идеальных аспектов и возвращающий иерархию аспектов по пользовательским отзывам.

# ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

Состав программной документации:

1) «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах». Программа и методика испытаний (ГОСТ 19.301 – 79\*);

2) «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах». Текст программы (ГОСТ 19.401 – 78\*);

3) «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах». Руководство оператора (ГОСТ 19.505 – 79\*);

4) «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах». Техническое задание (ГОСТ 19.201 – 78);

5) «Построение иерархии аспектов по пользовательским отзывам об электронных устройствах». Пояснительная записка (ГОСТ 19.404 – 79).

# СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1. Требования к техническим средствам

Для надежной и бесперебойной работы методов рекомендуется следующий состав аппаратурных средств:

1. персональный компьютер, оснащенный процессором с тактовой частотой 400 MГц и выше или аналогичный;
2. 128 ГБ оперативной памяти или больше (в связи с большими объемами анализируемых данных);

5.2. Требования к программным средствам

Для работы программы необходим следующий состав программных средств:

1. Установленный Python 2.7 или Python 3.2+.

5.3. Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

1. проверка требований к программной документации;
2. проверка требований к функциональным характеристикам.

5.4. Условия проведения испытаний

5.4.1. Требования к численности и квалификации персонала

Минимальное количество персонала, требуемого для работы программы, должно составлять не менее 1 штатной единицы:

1. конечный пользователь – человек, заинтересованный в построении иерархии.

Для работы с данной программой конечный пользователь должен:

1. обладать практическими навыками работы с программами в Python;
2. иметь базовое понимание построения иерархий.

# МЕТОДЫ ИСПЫТАНИЙ

Испытания представляют собой процесс установления соответствия программы заданным требованиям и программной документации.

6.1. Испытание выполнения требований к программной документации

Состав программной документации проверяется визуально, проверяется наличие всех подписей. Также визуально проверяется соответствие документации требованиям ГОСТ.

**6.2. Испытание выполнения требований к функциональным характеристикам**

**6.2.1. Метод парсер отзывов**

Для получения отзывов об электронных устройствах требуется в метод парсер передать адрес веб страницы, с которой будет начат парсинг, а также объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: url = “ulmart.ru“

In[2]: parser.process(url, db)

**6.2.2. Методы характеристик для вычисления семантического расстояния**

**6.2.2.1. PMI**

Для расчета значения PMI review и PMI sentence требуется установить библиотеки numpy и sklearn и подключить их с помощью команды import. В метод расчета PMI review передается корпус отзывов и корпус аспектов, в метод расчета PMI sentence передается корпус предложений из отзывов и корпус аспектов, а также в обоих случаях объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: import numpy as np  
   In[2]: from sklearn.feature\_extraction.text import CountVectorizer

In[3]: corpus\_review = [набор отзывов]

In[4]: corpus\_aspects = [набор аспектов]

In[5]: pmi.calculate\_pmi(corpus\_review, corspus\_aspects, db)

1. In[1]: import numpy as np  
   In[2]: from sklearn.feature\_extraction.text import CountVectorizer

In[3]: corpus\_sentence = [набор предложений]

In[4]: corpus\_aspects = [набор аспектов]

In[5]: pmi.calculate\_pmi(corpus\_ sentence, corspus\_aspects, db)

**6.2.2.2. Context**

Для расчета значения Context local и Context global требуется установить библиотеки numpy, scipy и sklearn и подключить их с помощью команды import. В метод расчета Context local передается корпус наборов 4-словных контекстов и корпус аспектов, в метод расчета Context global передается корпус отзывов и корпус аспектов, а также в обоих случаях объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: import numpy as np  
   In[2]: from sklearn.feature\_extraction.text import CountVectorizer

In[3]:from scipy import stats

In[4]: corpus\_context\_local = [предвычисленный набор контекстов для всех аспектов]

In[5]: corpus\_aspects = [набор аспектов]

In[6]: context.local\_context(corpus\_context\_local, corspus\_aspects, db)

1. In[1]: import numpy as np  
   In[2]: from sklearn.feature\_extraction.text import CountVectorizer

In[3]:from scipy import stats

In[4]: corpus\_context\_global = [предвычисленный набор отзывов, где встречается каждый из аспектов]

In[5]: corpus\_aspects = [набор аспектов]

In[6]: context.global\_context(corpus\_context\_global, corspus\_aspects, db)

**6.2.2.3. Lexical**

Для расчета значения Lexical в метод расчета передается корпус аспектов, а также объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: corpus\_aspects = [набор аспектов]

In[2]: lexical.process (corspus\_aspects, db)

**6.2.2.4. Syntactic**

Для расчета значения Syntactic требуется установить библиотеки numpy и sklearn и подключить их с помощью команды import. В метод расчета передается корпус аспектов, набор предвычисленных синтаксических деревьев предложений, корпус предложений, а также объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: import numpy as np  
   In[2]: from sklearn.feature\_extraction.text import CountVectorizer

In[3]: corpus\_sentence = [набор предложений]

In[4]: corpus\_syntactic = [набор предвычисленных синтаксических деревьев]

In[5]: corpus\_aspects = [набор аспектов]

In[6]: syntactic.process(corpus\_aspects, corpus\_syntactic, corpus\_sentence, db)

**6.2.3. Метод вычисления семантического расстояния**

Для расчета сематического расстояния требуется установить библиотеку numpy и подключить ее с помощью команды import. В метод расчета передается корпус характеристик для каждой пары аспектов и корпус аспектов, а также объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: import numpy as np

In[2]: corpus\_features = [предвычисленный набор характеристик для каждой пары аспектов]

In[3]: corpus\_aspects = [набор аспектов]

In[4]: semantic\_distance.process\_semantic\_distance\_learning(corpus\_aspects, corpus\_features, db)

**6.2.4. Метод построения иерархии**

Для построения иерархии аспектов по пользовательским отзывам об электронных устройствах требуется в метод расчета передать корпус идеальных аспектов, среднее значение семантичсекой дистанции для них, корпус аспектов и корпус семантичсеких расстояний для них, а также объект класса DB для возможности сохранения вычисленных результатов в базу данных.

1. In[1]: corspus\_ideal\_aspects = [корпус идеальных аспектов]

In[2]: avg\_distance = среднее значение семантичсекой дистанции для идеальных аспектов

In[3]: corpus\_aspects = [набор аспектов]

In[4]: corspus\_semantic\_distance = [корпус семантичсеких расстояний для аспектов]

In[5]: hierarchy.build\_hierarchy(corspus\_ideal\_aspects, avg\_distance, corpus\_aspects, corspus\_semantic\_distance, db)

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