|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **НПОУ «ЯКУТСКИЙ КОЛЛЕДЖ ИННОВАЦИОННЫХ ТЕХНОЛОГИЙ»** | | | | | | | |
| (наименование образовательной организации) | | | | | | | | | |
|  | | | | | | | | | |
|  | **ОТДЕЛЕНИЕ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ** | | | | | | |
| (наименование структурного подразделения (кафедра / отделение)) | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
| **ОТЧЕТ** | | | | | | | | | |
| **по Языкам программирования** | | | | | | | | | |
| Веб-парсинг сайта Озон для сбора отзывов с использованием семантического анализа текста | | | | | | | | | |
| (наименование типа практики) | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | | |  | Выполнил: | | | | | |
|  | | |  | обучающиеся | | КИСП-23 | группы | | |
|  | | |  | Варламов Никита Андреевич  Исаков Илья Игоревич  Левин Артем Данилович  Огонерова Сардаана Васильевна  Сафаргалеев Владимир Викторович  Яковлева Евгения Николаевна | | | | | |
|  | | |  | (фамилия, имя, отчество (при наличии)) | | | | | |
|  | | |  | \_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_ 20\_\_\_ г. | | | | | |
|  | | |  | (личная подпись, дата) | | | | | |
|  | | |  |  | | | | | |
|  | | |  |  | | | | | |
| Проверил руководитель практики | | |  | Проверил преподаватель: | | | | | |
|  | | |  |  | | | | | |
| (уч. степень, уч. звание, должность) | | |  | (уч. степень, уч. звание, должность) | | | | | |
|  | | |  | Федоров Дьулуур Андрианович | | | | | |
| (фамилия, имя, отчество (при наличии)) | | |  | (фамилия, имя, отчество (при наличии)) | | | | | |
| Оценка | |  |  | Оценка |  | | | | |
| \_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_ 20\_\_\_ г. | | |  | \_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_ 20\_\_\_ г. | | | | | |
| (личная подпись, дата) | | |  | (личная подпись, дата) | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
|  | | | | | | | | | |
| Якутск, 2025 | | | | | | | | | |

Содержание

[ВВЕДЕНИЕ 3](#_Toc195354893)

[1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ 4](#_Toc195354894)

[1.1. Терминология по проекту и глоссарий 4](#_Toc195354895)

[1.2. Распределение ролей и работы 6](#_Toc195354896)

[1.3. Стек технологий 7](#_Toc195354897)

[2. ПРАКТИЧЕСКАЯ ЧАСТЬ 8](#_Toc195354898)

[ЗАКЛЮЧЕНИЕ 9](#_Toc195354899)

ВВЕДЕНИЕ

Ozon является одним из крупнейших маркетплейсов на территории СНГ, предлагающим широкий выбор товаров в различных категориях. Удобная система поиска и интуитивно понятный личный кабинет позволяют пользователям легко находить нужные товары и отслеживать свои покупки.

Актуальность: Ozon - крупнейший маркетплейс на территории СНГ. Отзывы являются неотъемлемой частью любого маркетплейса. Они помогают, более детально изучить товар и подтолкнуть к его покупке или избежать некачественного товара.

Семантический анализ текста — это процесс оценки содержательности текста и определения его значения. Этот метод анализа помогает понять, какие ключевые идеи и смыслы заложены в тексте.

Объектом исследования процесс веб-парсинга сайта Озон для сбора отзывов с использованием семантического анализа текста, а методом исследования -  веб-парсинг и семантический анализ.

Целью проекта является создание веб-парсера для сбора отзывов с платформы Ozon для анализа тональности.

Задачи:

* Подготовка
* Парсинг страниц с отзывами
* Семантический анализ текста
* Визуализация результатов

1. ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

* 1. Терминология по проекту и глоссарий

Веб-парсинг (или парсинг данных) — это процесс автоматизированного извлечения информации с веб-сайтов. Этот метод позволяет собирать данные, которые обычно представлены в виде текста, изображений и других форматов, преобразуя их в структурированный вид для дальнейшего анализа или использования.

Для выполнения веб-парсинга используются специальные программы или скрипты, которые имитируют действия пользователя, загружают страницы сайта, анализируют HTML-код и извлекают нужные данные. Веб-парсинг может применяться для различных целей, таких как мониторинг цен, сбор новостей, создание баз данных и многое другое.

Для начала необходимо получить доступ к содержимому веб-страницы. Это делается с помощью HTTP-запросов. Популярные библиотеки для этого:

Python:

* Requests: самая популярная библиотека для отправки HTTP-запросов.
* Urllib: встроенная библиотека для работы с URL.
* Httpx: Современная асинхронная библиотека для работы с HTTP-запросами.
* Aiohttp: Асинхронная библиотека для работы с HTTP-запросами, которая использует asyncio.

Если же вы работаете над высоконагруженным проектом, где важна скорость и возможность параллельного выполнения запросов, стоит рассмотреть использование httpx или aiohttp.

Методы парсинга:

Ручной парсинг - это самый простой метод, когда вы вручную копируете данные с сайта. Подходит только для небольших объемов информации и не подходит для автоматизации.

Автоматический парсинг с использованием регулярных выражений

Регулярные выражения позволяют находить и извлекать текстовые шаблоны из HTML-кода страницы. Этот метод эффективен при работе с простыми структурами данных, но становится сложным при обработке сложных веб-страниц.

Парсинг с помощью CSS-селекторов

Многие библиотеки и фреймворки поддерживают работу с CSS-селекторами, что позволяет легко выбирать нужные элементы на странице. Например, document.querySelector в JavaScript или функции селекции в библиотеках вроде BeautifulSoup.

XPath

XPath – это язык запросов к XML-документам, который также используется для работы с HTML. С его помощью можно извлекать информацию из структурированных документов, включая сложные структуры страниц.

Если сайт предоставляет API, то использование этого метода будет наиболее предпочтительным, так как он обеспечивает доступ к данным напрямую без необходимости парсить HTML-код.

Терминология:

* Семантика: наука о значениях знаков и символов.
* Значимость: конкретные смыслы отдельных элементов текста.
* Контекст: окружающая ситуация, влияющая на понимание значения.
* Антецедент: исходный элемент, от которого зависит другое слово или выражение.
* Консистенция: согласованность смыслов внутри текста.
* Энтитизация: выделение сущностей (объектов, лиц, мест и др.).
* Инференция: вывод новых фактов из имеющихся данных.
* Кореференция: установление связи между разными частями текста, относящимися к одному объекту.
* Омонимия: наличие разных значений одного слова.
* Гипонимия/гиперонимия: отношения типа "род-вид" ("яблоко" гипоним "фрукта").
* Синтаксичность: взаимосвязь синтаксиса и семантики.
  1. Распределение ролей и работы

Проект был распределён среди участников команды следующим образом:

Левин Артем Сафаргалеев Владимир занимались бэкэнд разработкой:

Класс OzonParser

1. get\_page

Назначение: Загрузка HTML-страницы по указанному URL.

Параметры:

url: Адрес страницы для загрузки

params: Дополнительные параметры запроса (опционально)

Возвращает: HTML-код страницы или None при ошибке

1. Сделан метод parse\_search\_result

Назначение: Поиск товаров по ключевому слову и парсинг результатов.

Параметры:

* query: Поисковый запрос
* pages: Количество страниц результатов для обработки (по умолчанию 1)

Возвращает: Список словарей с информацией о товарах (название, бренд, цена, рейтинг, URL и т.д.)

Особенности:

* Обрабатывает указанное количество страниц результатов
* Использует BeautifulSoup для парсинга HTML
* Добавляет задержку между запросами (time.sleep(1))
* Собирает основную информацию о каждом товаре

1. Сделан метод parse\_product\_page

Назначение: Парсинг детальной страницы товара.

Параметры:

product\_url: URL страницы товара

Возвращает: Словарь с подробной информацией о товаре (название, бренд, цены, рейтинг, отзывы, характеристики, описание)

Особенности:

* Извлекает больше данных, чем parse\_search\_results
* Обрабатывает блоки с характеристиками и описанием товара

1. save\_to\_csv

Назначение: Сохранение данных в CSV-файл.

Параметры:

data: Данные для сохранения (список словарей)

filename: Имя файла для сохранения

Особенности:

* Использует pandas для создания DataFrame
* Сохраняет с кодировкой UTF-8 с BOM (utf-8-sig)

1. save\_to\_json

Назначение: Сохранение данных в JSON-файл.

Параметры:

data: Данные для сохранения (словарь)

filename: Имя файла для сохранения

Особенности:

Сохраняет с отступами для удобного чтения

Поддерживает Unicode-символы (ensure\_ascii=False)

Функция Main()

Назначение: Точка входа в программу, предоставляет пользовательский интерфейс для выбора действий.

Функционал:

* Создает экземпляр парсера
* Предлагает пользователю выбрать действие:
* Парсинг результатов поиска
* Парсинг детальной страницы товара
* В зависимости от выбора запрашивает дополнительные параметры
* Сохраняет результаты в соответствующий файл (CSV или JSON)

Исаков Илья занимался разработкой фронтэнд.

Создание телеграмм бота

Настройка

Создание логики работы бота

Яковлева Евгения занималась созданием базы данных:

Создание базы данных

Подключение database mail

Настройка базы данных

Огонерова Сардаана занималась созданием базы данных:

Создание таблиц

Настройка таблиц

Создание и оптимизация SQL запросов

Варламов Никита занимался созданием ИИ:

Создание генеративной речевой модели на базе YandexGPT

Создание логики работы.

1.3 Стек технологий

Основные технологии:

* Python - основной язык программирования
* Requests - библиотека для выполнения HTTP-запросов
* BeautifulSoup4 - библиотека для парсинга HTML/XML
* Pandas - библиотека для работы с данными и экспорта в CSV
* Fake-useragent - генерация случайных User-Agent строк

Вспомогательные библиотеки:

* Time - стандартная библиотека Python для работы со временем (используется для задержек между запросами)
* JSON - стандартная библиотека Python для работы с JSON форматом
* Random - может использоваться для генерации случайных задержек (хотя в текущей реализации не используется)

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

ЗАКЛЮЧЕНИЕ

Использование результатов анализа отзывов открывает перед бизнесом широкие возможности для улучшения всех аспектов своей деятельности – от разработки новых продуктов до оптимизации внутренних процессов. Главное – внимательно слушать своих клиентов и своевременно реагировать на их обратную связь.