Структура КП ПРИМЕР 1

Оглавление

Введение

Глава 1 Обзор и анализ методов и алгоритмов интеллектуального анализа текстовых данных

1.1. Метод токенизации

1.2. Частота термина в документе (TF-IDF)

1.3. Методы стемминга и лемматизации

1.4. Стоп-листинг

Глава 2 Обзор и анализ алгоритмов интеллектуального анализа текстовых данных

2.1. Алгоритмы токенизации

2.2. Алгоритм TF-IDF

2.3. Алгоритм лемматизации WordNet

2.4. Алгоритмы стоп-листинга

Глава 3 Разработка и реализация интеллектуального анализа текстовых данных

3.1. Выбор среды разработки программы

3.1.1. Интегрированная среда разработки Python

3.2. Реализация и тестирование программы

Заключение

Список используемой литературы

Введение

В настоящее многие организации осознали пользу от внедрения аналитических инструментов для поддержки принятия решений. Одним из новых направлений в этой области является текстовая аналитика.

Текстовая аналитика — это автоматизированный процесс преобразования больших объемов неструктурированного текста в количественные данные для выявления идей, тенденций и закономерностей. В сочетании с инструментами визуализации данных этот метод позволяет компаниям понять суть цифр и принимать более обоснованные решения [1].

Как показывает практика текстовой аналитики, наиболее эффективными средствами повышения качества анализа текстов являются методы и алгоритмы интеллектуального анализа данных. Применение алгоритмов интеллектуального анализа текстовых данных представляет актуальность и научно-практический интерес. Объектом исследования бакалаврской работы являются методы и алгоритмы интеллектуального анализа текстовых данных.

Предметом исследования курсового проекта является применение алгоритмов интеллектуального анализа текстовых данных.

Целью курсового проекта является исследование особенностей практического применения алгоритмов интеллектуального анализа для повышения качества анализа текстовых данных в образовании.

Для достижения данной цели необходимо выполнить следующие задачи:

− провести анализ методов и алгоритмов интеллектуального анализа текстовых данных;

− исследовать особенности применения алгоритмов интеллектуального анализа текстовых данных в различных прикладных задачах;

− разработать и протестировать программу, реализующую алгоритмы интеллектуального анализа текстовых данных.

Методы исследования – текстовая аналитика, Text Mining, методы и технологии проектирования программного обеспечения.

Практическая значимость курсового проекта заключается в разработке программы, реализующей эффективные алгоритмы интеллектуального анализа текстовых данных.

Данная работа состоит из введения, трех глав, заключения и списка используемой литературы.

Первая глава посвящена обзору и анализу методов и алгоритмов интеллектуального анализа текстовых данных.

Во второй главе проанализированы особенности применения алгоритмов интеллектуального анализа текстовых данных в различных прикладных задачах.

В третьей главе описан процесс разработки и тестирования программы, реализующей алгоритмы интеллектуального анализа текстовых данных.

Глава 1 Обзор и анализ методов и алгоритмов интеллектуального анализа текстовых данных

Следует отметить, что методы интеллектуального анализа текстовых данных относятся к области Natural Language Processing (NLP) или обработке естественного языка.

NLP в широком смысле определяется как автоматическая обработка естественного языка, такого как речь и текст, с помощью программного обеспечения.

Обработка естественного позволяет машинам разбирать и интерпретировать человеческий язык.

NLP лежит в основе инструментов, которые мы используем каждый день - от программного обеспечения для перевода, чат-ботов, фильтров спама и поисковых систем до программного обеспечения для исправления грамматики, голосовых помощников и инструментов для мониторинга социальных сетей (рисунок 1) [17].

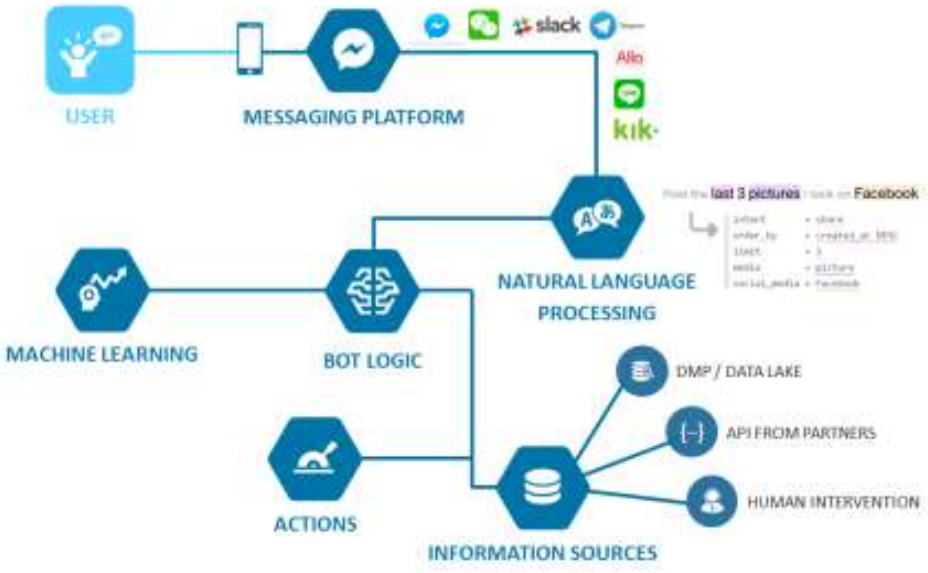

Рисунок 1 – Области применения NLP

Глава 2 Обзор и анализ алгоритмов интеллектуального анализа текстовых данных

Рассмотрим и проанализируем алгоритмы интеллектуального анализа текстовых данных, построенных на основе методов, описанных в главе 1.

2.1 Алгоритмы токенизации

Рассмотрим токенизацию на примере алгоритмов обработки подслов.

«Подслово - это некоторая строгая последовательность символов слова, начинающаяся в определенной позиции. Для ее получения берется часть слова, представляющая собой последовательность символов определенной длины. Если представить слово набором букв, то подслово представляет собой содержимое некоторого окна заданной длины, наложенного на это слово.

Корпусом называют собрание текстов или фрагментов текстов в электронной форме, отобранных в соответствии с внешними критериями, чтобы наиболее полно представлять язык или вариацию языка. Кроме текстовых данных корпус составляет программное обеспечение системы управления и анализа текстов» [22].

Рассмотрим алгоритм кодирования пар байтов (Byte Pair Encoding, BPE).

Пошаговое представление алгоритма BPE имеет вид:

Шаг 1. Подготовьте достаточно большие обучающие данные (например,

Глава 3 Разработка программы интеллектуального анализа текстовых данных

Для разработки программы интеллектуального анализа текстовых данных используем язык программирования Python и технологию Integrated Development Environment (IDE).

3.1 Выбор среды разработки программы

IDE – интегрированная среда разработки представляет собой многофункциональную программу, которую можно использовать для различных аспектов разработки программного обеспечения.

Интегрированная среда разработки позволяет программистам объединять различные задачи написания компьютерной программы.

Интегрированные среды разработки повышают продуктивность программиста за счет объединения общих действий по написанию программного обеспечения в одном приложении: редактирование исходного кода, создание исполняемых файлов и отладка.

В состав типовой интегрированной среды разработки входят:

- текстовый редактор;
- транслятор – компилятор или интерпретатор;
- средства автоматизации сборки;
- отладчик.

Рассмотрим функциональные и архитектурные особенности популярных интегрированных сред разработки.

Список используемой литературы

1. Бахтин А.В. Алгоритмы извлечения из неструктурированных текстовых источников метаинформации о научно-технических конференциях. М: МГУ [Электронный ресурс]. URL: https://www.hse.ru/data/2015/06/07/1097438594/presentation_cfp.pdf (дата обращения: 10.06.2021).

2. Библиотека NTLK [Электронный ресурс]. URL: http://www.nltk.org/ (дата обращения: 10.06.2021).

3. ВКонтакте опубликовали библиотеку для предобработки текстовых данных [Электронный ресурс]. URL: https://neurohive.io/ru/novosti/vkontakte-opublikovali-biblioteku-dlya-predobrabotki-tekstovyh-dannyh/ (дата обращения: 10.06.2021).

4. Кластеризация и классификация больших текстовых данных с помощью машинного обучения на Java [Электронный ресурс]. URL: https://itnan.ru/post.php?c=1&p=529548 (дата обращения: 10.06.2021).

5. Краткое руководство. Знакомство с интегрированной средой разработки Visual Studio [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/visualstudio/ide/quickstart-ide-orientation?view=vs-2019 (дата обращения: 10.06.2021).

6. Ле Мань Ха. Оптимизация алгоритма KNN для классификации // ТРУДЫ МФТИ. 2016. Том 8, № 1. С. 92-94.

7. Левенштейн В.И. Двоичные коды с исправлением выпадений, вставок и замещений символов // Докл. АН СССР. 1965. 163 (4). С. 845–848.

8. Метод TF-IDF [Электронный ресурс]. URL: https://ru.wikipedia.org/wiki/TF-IDF (дата обращения: 10.06.2021).

9. Отраднов К.К., Раев В.К. Экспериментальное исследование эффективности методик векторизации текстовых документов и алгоритмов их кластеризации. Вестник РГРТУ. 2018. № 64. С. 74-82.