МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»

Кафедра информатики и программирования

Создание программного обеспечения для анализа неструктурированного текста

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование информационных систем
факультета компьютерных наук и информационных технологий
Косяна Арама Араиковича

Научный руководитель:
Старший преподаватель _____ Е.Е. Лапшева

подпись, дата

Зав. кафедрой:
к. ф.-м. н., доцент _____ М.В. Огнева

подпись, дата

Саратов 2020

ВВЕДЕНИЕ

Интеллектуальный анализ текста или контент-анализ – это набор методов для автоматического извлечения полезной информации из текстов, написанных на естественном языке. Исследователи и разработчики могут использовать эти методы для сбора разнообразных и неорганизованных данных в структурированную форму. В этом процессе документы дезинтегрируются для беспроблемного управления фрагментами данных, проще говоря: неструктурированный текст преобразуется в структурированные данные. Так извлеченные из неструктурированной информации данные должны быть подготовлены к последующему анализу. При этом важно отметить, что эффективное и точное преобразование неструктурированных данных в процессе извлечения информации улучшает последующий анализ данных. В настоящее время существует множество методов анализа, созданных специально для разных типов информации: текста, изображения, аудио и видео.

В последние годы успехи в развитии технологий способствовали стремительному росту объемов данных. Объем, разнообразие, а также скорость обработки больших данных изменили парадигму вычислительных возможностей систем. По оценке компании IBM, ежедневно генерируется более 2,5 квинтиллионов байтов информации. На фоне этих статистических показателей также можно ожидать, что доля неструктурированных данных из различных источников увеличится до 90% в ближайшие несколько лет. Согласно прогнозу компании IDC, неструктурированные данные составят 95% от глобального объема информации уже в 2020 году с прогнозируемым годовым ростом в 65% [1].

Из-за значительных объемов и сложности неструктурированных данных, очень трудно извлекать полезную информацию из материалов различного типа. Интеграция данных, хранящихся как в структурированных, так и в неструктурированных форматах, может значительно повысить ценность информации. Анализ и обработка неструктурированных данных, их

форматирование и слияние с традиционными структурированными данными обеспечивает более глубокое корпоративное понимание для лица, принимающего решение.

Важно отметить, что неструктурированные данные состоят из произвольной формы текста, таких как текстовые документы, электронные письма, веб-страницы, аудио- и видеопотоки, изображения. Главным препятствием для извлечения информации из такого неструктурированного документа является его неорганизованное, неоднозначное и свернутое содержание. Это требует более высокого уровня предварительной обработки, чем типичные методы предварительной обработки, принятые для структурированных и полуструктурированных документов.

Таким образом, неструктурированные данные представляют значительные проблемы для исследователей данных, часто требующие чрезмерного количества времени для структурирования и подготовки данных к анализу. Неструктурированные источники данных очень велики по объему.

Цель настоящей работы: разработать программное обеспечение на основе PyQt5 для анализа неструктурированного текста. Для достижения поставленной цели необходимо решить следующие **задачи:**

- Исследовать методы анализа неструктурированных данных. Сделать обзор существующих технологий.
- Исследовать методы анализа неструктурированного текста.
- Освоить технологии создания ПО на основе PyQt5.
- Изучить обработку неструктурированного текста с помощью библиотеки NLTK.
- Разработка ПО для анализа неструктурированного текста.

Методологические основы «Создание программного обеспечения для анализа неструктурированного текста» представлены в работах Сачкова В. Е., Гильмутдиновой Е. Ф., Матяша Е. Д., Акимова Д. А., также в работах зарубежных авторов: Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin.

Теоретическая значимость бакалаврской работы заключается в успешно осуществленном методе анализа неструктурированного текста. Полученные сведения могут быть использованы для дальнейшего изучения поставленного вопроса.

Практическая значимость бакалаврской работы заключается в том, что результаты анализа неструктурированного текста в дальнейшем можно применять в различных целях, например, для оптимизации сайтов. А также помогает SEO (Search Engine Optimization) – специалистам сократить время на создание и проработку текста.

Структура и объём работы. Бакалаврская работа состоит из введения, 4 разделов, заключения, списка использованных источников и 5 приложений. Общий объем работы – 72 страницы, из них 60 страниц – основное содержание, включая 17 рисунков и 1 таблицу, цифровой носитель в качестве приложения, список использованных источников информации – 19 наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Методы анализа неструктурированного текста» посвящен исследованию методам анализа неструктурированных данных.

Неструктурированные данные – информация, которая либо не имеет заранее определенной структуры данных, либо не организована в установленном порядке. Неструктурированные данные обычно представлены как набор символов, образующий некоторый текст, который может содержать кроме символов естественного алфавита, используемого для образования слов, и данные, записанные с помощью цифр (даты, числа и иные факты). Существуют следующие техники поиска закономерностей с целью интерпретировать неструктурированную информацию: интеллектуальный анализ данных (datamining), обработка естественного языка (Natural Language Processing) и интеллектуальный анализ текста [2].

Обработка естественного языка (Natural Language Processing, NLP) – это область искусственного интеллекта (ИИ), которая позволяет техническим устройствам понимать, интерпретировать и манипулировать человеческим языком. Основной задачей NLP является разделение текста на элементарные частицы (слова, предложения, абзацы).

Существует большое разнообразие задач обработки естественного языка:

1. Поиск фрагментов текста.
2. Поиск предложений (Sentence Boundary Disambiguation, SBD).
3. Поиск именованных объектов (Named entity recognition, NER).
4. Определение частей речи (Parts of speech, POS).
5. Классификация текстов и документов.
6. Выделение взаимоотношений. [3].

Поиск отдельных слов и разделение исходных данных на более мелкие фрагменты является основной задачей при обработке текста. Токенизация — это процесс разделения текста на отдельные мелкие фрагменты (слова, словосочетания, предложения).

Разделение сырой строки на значимые лексемы называется процессом лексемизация. Сложность токенизации варьируется в зависимости от необходимости применения NLP, и сложности самого языка.

Наиболее известным и часто используемым направлением обработки текста является это его перевод с одного естественного языка на другой. Одной из наиболее продвинутых методик, используемых в настоящее время для достижения правильного перевода, является использование нейронных сетей типа “Seq2Seq”, а также “Word2Vec”.

Структура нейронной сети возникла в программировании по аналогии с биологическими объектами. Искусственные нейронные сети – это система соединённых и взаимодействующих между собой нейронов (простых процессоров). Алгоритм работы таких процессоров крайне прост.

В работе рассматриваются рекуррентные нейронные сети, которые позволяют сохранять информацию во времени, «запоминая» определенные данные, также СНС (CNN) - сверточные нейронные сети в большинстве случаев предназначены для распознавания объектов и образов на картинках, классификации изображений, отделение особенностей и сжатии данных [4].

Для анализа текста используется посимвольный подход и подход с использованием кодирования слов. Каждый подход имеет свои особенности.

Обычно тексты содержат разные грамматические формы одного и того же слова, изменяемые по падежам, склонениям и временам. Лемматизация и стемминг имеют цель привести все встречающиеся словоформы к одной, нормальной словарной форме (именительный падеж у существительных, инфинитив у глаголов и т.д.).

Лемматизация и стемминг – это частные случаи нормализации и они отличаются способом преобразования слов к основе.

Стемминг – это эвристический процесс, который отрезает «лишнее» от корня слов (на данном этапе происходит удаление суффиксов и окончаний), часто это приводит к потере словообразовательных суффиксов.

Лемматизация – это более сложный и достаточно тонкий эвристический процесс, который использует словарь и морфологический анализ для приведения слова к его канонической форме – лемме [5].

Обрабатывать неструктурированный текст позволяют такие программные продукты как Stanford Core NLP Natural, Language Tool kit, NLTK, Scikitlearn, Tensorflow. Каждый из них имеет свои особенности работы.

Таким образом, NLP является наиболее интересным для многих организаций. Данные технологии могут дать большой спектр ценных идей и решений для разработки новых приложений. Они способны решить языковые проблемы, с которыми могут столкнуться потребители при взаимодействии с продуктом. Исходя из данных причин, и в данной работе также будет использован метод Natural language processing (NLP) с библиотекой NLTK.

Второй раздел «Обработка неструктурированного текста с помощью библиотеки NLTK» посвящен анализу библиотеки NLTK.

Natural Language Tool Kit (NLTK) — это комплексная библиотека Python для обработки естественного языка и анализа текста. Первоначально разработанная для обучения, она была принята в промышленности для исследований и разработок благодаря своей полезности и широте охвата. NLTK часто используется для быстрого создания прототипов программ обработки текста и может даже использоваться в производственных приложениях.

Модуль NLTK можно установить точно так же, как и любой другой модуль Python, либо непосредственно загрузив пакет с веб-сайта NLTK, либо используя инсталляторы сторонних разработчиков с ключевым словом «nltk» [6]. NLTK отлично подходит для получения статистической информации об общем количестве слов, количестве уникальных слов, частоте встречаемости слов в тех или иных разделах текста.

В NLTK есть предустановленный список стоп-слов. Перед первым использованием вам понадобится его скачать: nltk.download (“stopwords”).

Для дополнительного фильтрования текста используются регулярные выражения. Регулярные выражения – это мощный инструмент, с его помощью можно создавать гораздо более сложные шаблоны.

Таким образом, NLTK (Natural Language Tool kit) – является одной из ведущих платформ для создания NLP-программ на языке программирования Python, включающий различные методы.

Третий раздел «Технология создания ПО на основе PyQT5» посвящен программе PyQT5, его компонентам и виджетам.

PyQt является привязкой Python для инструментария Qt, который ещё и функционирует как кросс-платформенная среда разработки приложений. Сам Qt написан на популярной среде C ++. Используя его на Python, можно гораздо быстрее создавать приложения, не жертвуя при этом значительной скоростью C++.PyQt5 совместим с Windows, Unix, Linux, macOS.

На данный момент самой актуальной версией PyQt является версия PyQt5, которая не имеет обратной совместимости со старыми модулями предыдущих версий.

Существует два способа установки PyQt:

- Сборка и установка из исходного кода;
- Использование файлов Wheel [7].

PyQt содержит большое количество виджетов для создания различных приложений с помощью графического интерфейса. Но в новой версии PyQt5 были произведены реорганизации классов в другие модули и ревизии в лицензиях.

PyQt обладает удобным механизмом управления компоновкой. Его можно использовать для создания расширенных пользовательских интерфейсов при разработке приложений.

При создании приложений удобно использовать встроенные темы, которые поставляются с PyQt5. Для установки определенной темы следует использовать метод setStyle(), вызываемый в экземпляре QApplication [8].

Таким образом, программное обеспечение будет разрабатываться на основе PyQt5 для анализа неструктурированного текста, так как используя его на Python, можно гораздо быстрее создавать приложения, не жертвуя при этом значительной скоростью C++.

В четвертом разделе «Создание ПО для анализа неструктурированного текста» показан алгоритм создания ПО для получения плотности ключевого слова, преобразование программы в desktop-приложение, используя Pyinstaller, а также примеры использования полученной программы.

В работе описывается последовательность создания программы для получения плотности ключевого слова английского неструктурированного текста:

- Получили отформатированный текст из Html-документов с помощью Beautiful Soup и requests
- Использовали библиотеку NLTK для структурирования текста
- Избавились от стоп-слов
- Создали график с часто-встречающимися словами
- Определили плотность ключевых слов в процентах и показали на графике.

Плотностью ключевых слов называют показатель частоты повторения ключевиков в тексте. Рассчитывается, как процентное отношение количества ключевых слов к общему числу слов в тексте. Одним из основных параметров является плотность ключевых слов, с его помощью поисковые системы вычисляют уровень соответствия запрашиваемой информации в поиске.

По статистике показатель оптимальной плотности ключевых слов составляет примерно 3-8% от всего текста не является точной. На самом деле все индивидуально. Количество ключевых слов на странице значительно зависит от объема текста, положения страницы на сайте, и, конечно,

тематики статьи, набора различных списков стоп-слов, которые использует каждый анализатор плотности ключевых слов.

Программный продукт не обходится без графического пользовательского интерфейса. Qt Designer– инструмент для проектирования и создания графических пользовательских интерфейсов (GUI) из компонентов Qt. Можно создать и настроить свои виджеты или диалоги в режиме «что видишь, то и получишь» (what-you-see-is-what-you-get, WYSIWYG), и проверить их, используя разные стили и разрешающие способности [8].

Для того, чтобы преобразовать программу в desktop-приложение, используем продукт Pyinstaller. Pyinstaller собирает Python-приложение и все зависимости в один пакет. В данном случае пользователь может запускать приложение без установки интерпретатора Python или иных модулей. Данный способ предпочтительнее, т.к. приложение становится более доступным и не ориентировано на специализированное программное обеспечение. Pyinstaller поддерживает Python 2.7 и Python 3.3+ и некоторые библиотеки: numpy, PyQt, wxPython, Django и другие. Может быть установлен с использованием pip [9].

После этого можно рассмотреть примеры использования полученного программного продукта. Программный продукт позволяет анализировать информацию по url ссылке, добавление документа форматом txt либо вставкой обычного текста, результатом анализа является таблица с процентами плотности ключевых слов, которую можно сохранить в формате xls. Более того предусмотрена кнопка График, благодаря чему появляется график с возможностью масштабирования и сохранения этого графика в виде png файла на рабочий стол

Таким образом, многообразие систем автоматической обработки неструктурированных текстов сегодня вызывает необходимость их систематизации и классификации с целью упрощения выбора решения, наиболее адекватного для конкретной задачи.

ЗАКЛЮЧЕНИЕ

В данной работе были изучены методы анализа неструктурированного текста, сделан обзор существующих технологий для анализа неструктурированного текста и неструктурированных данных. Разработано программное обеспечение (ПО) для анализа неструктурированного текста на основе набора расширений PyQt5 графического фреймворка Qt на языке программирования Python, способный взаимодействовать с пользователем на операционной системы Windows. Для обработки неструктурированного текста на языке программирования Python был взят и изучен открытый набор библиотек NLTK.

Результаты анализа неструктурированного текста в дальнейшем можно применять в различных целях, например, для оптимизации сайтов. А также помогает SEO (Search Engine Optimization) – специалистам сократить время на создание и проработку текста

Основные источники информации:

1. IBM, "IBM Study: Digital Era Transforming CMO's Agenda, Revealing Gap in Readiness", IBM news release, October 11, 2011

2. Неиспользованный потенциал в неструктурированном тексте - Мэри Бет Мур, стратегический специалист по ИИ и языковой аналитике, SAS [электронный ресурс] // Режим доступа: https://www.sas.com/ru_ru/insights/articles/analytics/the-untapped-potential-in-unstructured-text.html

3. Сачков В. Е., Гильмутдинова Е. Ф., Матяш Е. Д., Акимов Д. А. Обработка и компьютерный анализ текста на естественных языках // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и Технические Науки. -2016. -№12. -С. 57-64

4. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, Illia Polosukhin Attention Is All You Need – ARXIV, Электронная версия печ. публикации arXiv:1706.03762, 06/2017 – PDF формат, версия 5 [электронный ресурс] // Режим доступа: https://arxiv.org/pdf/1706.03762.pdf

5. Анализ_тональности_текста [электронный ресурс] // Режим доступа: https://ru.wikipedia.org/wiki/Анализ_тональности_текста

6.Natural Language Tool kit[электронный ресурс] // Режим доступа: http://www.nltk.org/

7.About PyQt [электронный ресурс] // Режим доступа: https://wiki.python.org/moin/PyQt

8. QtDesigner Manual [электронный ресурс] // Режим доступа: https://doc.qt.io/qt-5/qtdesigner-manual.html

9. PyInstaller Manual [электронный ресурс] // Режим доступа: https://pyinstaller.readthedocs.io/en/stable/