МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра Математического и компьютерного моделирования

Построение информационной системы для анализа тенденций IT рынка с применением технологий Big Data и Data Mining

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 247 группы

направление 09.04.03 — Прикладная информатика

механико-математического факультета

Тураева Артема Фирузовича

Научный руководитель
профессор, д.э.н., профессор Л.В. Кальянов

Зав. кафедрой
зав. каф., д.ф.-м.н., доцент Ю.А. Блинков

Саратов 2022

Введение. Информационные технологии развиваются стремительным темпом. Почти все сферы деятельности человека связаны с обработкой поступающей информации извне. Интернет за последнее десятилетие обогатился огромным количеством данных, что привело к возникновению целому ряду задач по хранению и обработке неструктурированной, разрозненной информации. В конечном счёте появилась дисциплина решающая целый спектр такого рода задач - Big Data.

Сегодня всё большей популярностью пользуются методы из Data Mining, которые по своей сути можно охарактеризовать, как просеивания огромного количества сырого материала с целью нахождения закономерностей. Самыми значимыми на текущий момент являются методы из Text Mining. Данные методы находятся на стыке лингвистики и алгоритмистики. Это привносит ряд ограничений на использование стандартных методов и требуют более детального погружения, как в предметную область, так и в лексически смежные области. Все эти подходы сводятся к различным вычислительным способам обработки человеческих языков - NLP, которые в свою очередь могут быть реализованы с применением алгоритмов машинного обучения и нейронных сетей.

Целью работы является выявление тенденций IT рынка. Можно выделить 2 задачи для достижения поставленной цели:

1) Разработка эффективного программного обеспечения, позволяющего осуществлять сбор информации (веб краулер) и сохранение обработанных данных в БД;
2) Построение информационной системы для обработки и анализа собранных данных.

Структура магистерской работы. Данная работа включает 6 разделов:

1) Постановка задачи сбора информации из всемирной сети, анализ различных подходов к извлечению информации;
2) Описание систем хранения информации и преимущества выбора NoSql;
3) Рассматриваются технологии языка python, позволяющие писать производительные, расширяемые и бесперебойные системы сбора информации;

4) Описание технологий Big Data и источников данных;
5) Разбираются основные подходы и методы анализа текстовой информации;
6) Разработка системы анализа тенденций IT рынка.

Основное содержание работы.

Веб-граф описывает направленные ссылки между страницами Всемирной паутины. В общем случае граф состоит из нескольких вершин, некоторые пары из которых соединены между собой ребрами. В ориентированном графе, ребра имеют направление. Веб-граф является ориентированным графом, вершины которого соответствуют веб-страницам сети, а рёбра — связями между ними. При этом ребро направленное со страницы X в направлении страницы Y строится в случае, если на странице X есть гиперссылка на страницу Y.

Web crawler (Сетевой паук, краулер) — программа, являющаяся составной частью поисковой системы и предназначенная для перебора страниц интернета с целью занесения информации о них в базу данных.

Порядок обхода страниц, частота визитов, защита от зацикливания, а также критерии выделения значимой информации определяются алгоритмами информационного поиска.

Принцип обхода сайта.

В связи с тем, что сайт можно представить в виде графа, в таком случае возможно появление циклических ссылок (страницы ссылаются друг на друга). Для предотвращения подобных случаев в веб краулер заложена логика индексирования уже пройденных страниц, а для того, чтобы только ресурсы с определённым доменным именем также внесено дополнительная проверка доменного имени сайта, что позволило осуществлять парсинг страниц по html тегам и извлекать необходимую информацию с ресурса.

Алгоритм поиска в ширину систематически обходит все ребра графа для «открытия» всех вершин, достижимых из начального узла. Поиск в ширину имеет такое название потому, что в процессе обхода мы идём вширь, т.е. перед тем как приступить к поиску вершин на расстоянии k+1 от корневого узла, выполняется обход всех вершин на расстоянии k от него.

Algorithm 1 Модифицированный алгоритм BFS

Вход: начальный узел r; глубина обхода d

Выход: *get* запрос; распаренный *html*

```
1:  q, v_n, ctr = Queue(r), set(), 0
2:  while q do
3:      v = q.get()
4:      if v not in v_n and not is_visited(v) then
5:          v_n.add(v)
6:          req = self._get_timed_out_request(v, s_t=0.5)
7:          soup = BeautifulSoup(req.text, "lxml")
8:          yield req, soup
9:          if ctr < d then
10:             while Есть гиперссылки на странице (new_v) do
11:                 q.put(new_v)
12:                 ctr += 1
13:                 if ctr == d then
14:                     break
15:                 end if
16:             end while
17:         end if
18:     end if
19: end while
```

Системы хранения данных.

В данном разделе рассмотрены ключевые особенности хранения данных, описаны различия между SQL и NoSql базами данных.

Была выбрана документоориентированная база - MongoDB. Изучены ключевые особенности БД и представлен пример документа, сохранённой новостной статьи:

```
"_id": { "$oid": "5faa70cd7ff17d05f134d9cc" },
"url": "https://www.it-world.ru/it-news/market/183658.html",
"date": {"$date": {"$numberLong": "1649462400000"}},
"article": "...Новостная статья...",
```

```
"plain_body_text": "...Извлечённый текст <body>..."
```

Технологии языка python для построения Web Crawler.

Процесс извлечения необходимой информации с веб страницы может выглядеть следующим образом:

- Ввести URL-адрес в браузер.
- Подождать пока откроется веб страница.
- Найти необходимую информацию на странице.
- Сохранить эту информацию.
- Повторить все шаги для другой нужной страницы.

Все вышеперечисленные шаги можно автоматизировать и по итогу получится веб краулер (spider). Во время краулинга необходимо также осуществлять скрапинг для извлечения нужной информации из HTML страницы.

Внедрение многопоточности.

Для построения многопоточного crawling использовался пакет concurrent. Данный пакет предоставляет унифицированный интерфейс для решения задач межпоточной коммуникации и координации, что может быть реализовано посредством пулов потоков или процессов. Этот модуль содержит два класса ThreadPoolExecutor (использующий рабочие потоки) и ProcessPoolExecutor (использующий рабочие процессы), которые реализуют один и тот же абстрактный интерфейс: Executor.

Подготовка данных.

Прежде чем сохранить данные в базу данных их необходимо очистить от ненужной информации, которая будет занимать место в хранилище.

Главная очистка данных заключается в удалении html тегов извлечённых страниц. Для этой цели хорошо подходит модуль lxml, который способен очистить текст, от разметки, которая может быть частично правильной.

Реализация BeautifulSoup уже предоставляет возможность передавать настройки парсера в качестве параметра features.

Для удаления разного рода пробельных конструкций было написано регулярное выражение, чтобы заменить все вхождения подобных символов на обычный пробельный символ.

```
re.sub("\xa0|\\r|\\t|\\v", " ", text)
```

Сервис взаимодействия с БД.

Для того, чтобы подключить программу к NoSql базе нужно запустить локальный сервер MongoDB на своей машине и подключиться к localhost к порту по умолчанию с помощью драйвера для MongoDB - pymongo.

Логирование.

Во время выполнения программы могут происходить некоторые сбои и ошибки, чтобы их отследить лучший инструмент - логирование. Python предоставляет удобную библиотеку logging. Чтобы вывод логов был более информативным, его нужно настроить с помошью logging.Formatter, также с помощью настроек можно указать в какое место будут записываться логи - в файл или на консоль.

Тестирование.

Всемирная сеть динамически изменяется каждый день - некоторые сайты перестают работать или меняется их разметка и в итоге веб скрапер перестаёт выдавать нужный результат. Для того, чтобы проверить работоспособность программы было введено юнит тестирование (модуль unittest).

Обработка сбоев.

Во время работы программы возможны ряд критических сбоев, таких как отключение электропитания или сбои подключения сети интернет.

Чтобы обработать такого рода критические ошибки был введён механизм сериализации состояния краулер движков. Этот способ сохраняет список извлечённых страниц с сайта и номер текущей страницы обхода. При возникновении сбоев в работе программы, повторный запуск может быть осуществлён с предыдущего состояния (до сбоя).

Архитектура приложения.

При проектировании архитектуры приложения следует делать её как можно более гибкой и масштабируемой. В итоге было такое приложение, что при необходимости можно быстро добавить новый сайт для кроулинга, либо сменять базу данных при необходимости.

Результаты краулинга.

В итоге работы программы было собрано 19569 статей с новостных сайтов: `www.rbc.ru`, `www.cnews.ru`, `www.it-world.ru`, `https://3dnews.ru/`, `https://hi-tech.news`, `https://habr.com`. Текстовое содержание спарсенных стра-

ниц помещено в нереляционное хранилище MongoDB. Все url страниц были проиндексированы для дедубликации и однозначного соответствия ссылки на сайт с её содержимым.

Big Data.

Большие данные — обозначение структурированных и неструктурированных данных огромных объёмов и значительного многообразия.

Совокупность подходов и инструментов обработки больших данных, с точки зрения информационных технологий, включала средства параллельной обработки, такие как алгоритмы MapReduce с реализующими их проектами - Hadoop; шардированные хранилища, имеющие СУБД категории NoSQL и алгоритмы PageRank для выявления значимых источников данных.

Data Mining.

Интеллектуальный анализ текста, или text mining — автоматизация извлечения сведений из текстовых данных. Text Mining охватывает новые методы для выполнения семантического анализа текстов, информационного поиска и управления. Синонимом понятия Text Mining является KDT (Knowledge Discovering in Text - поиск или обнаружение знаний в тексте).

В ходе изучения Text Mining были выявлены основные методы и алгоритмы обработки информации, в частности для анализа текстов на русском языке необходимо применять эти методы с учётом языковых особенностей:

- *Токенизация* - позволяет разбить длинные участки текста на более мелкие.
- *Нормализация* - приведение текста к структурированному виду.
- *Стемминг* - приведение слова к его корню.
- *Лемматизация* - приведение слова к смысловой канонической форме слова.
- *Очистка стоп-слов* - избавление от всех слов, не несущих смысловой нагрузки.
- *Тематическое моделирование* - выявление основных тем (топиков) в документе.
- *Мешок слов* - определение количества вхождений слов в текст.
- *N-граммы* - определение количества вхождений слов из N элементов в текст.

- $TF - IDF$ - частотный метод, учитывающий частоту, с которой слово появляется в корпусе.
- *Анализ тональности текста* - класс методов контент-анализа для выявления эмоционально окрашенной лексики в тексте.
- *K-Means* - один из алгоритмов машинного обучения, решающий задачу кластеризации.

Платформы аналитики.

Были применены 2 системы аналитики: RapidMiner и KNIME.

В ходе составления аналитического процесса на RapidMiner были выявлены критические моменты, такие как переполнение памяти и ошибки в новых версиях программы.

KNIME в свою очередь предоставил удобны и гибкий функционал аналитики, по методу составления аналитического процесса обе программы работают одинаково, но KNIME лишён большинства недостатков RapidMiner.

Анализ данных с помощью KNIME.

Извлечение и подготовка данных.

Прежде всего необходимо загрузить данные из базы данных в программу, для этого в KNIME предусмотрен ряд расширений, которые расширяют базовый функционал системы.

Было установлено соединение с сервером MongoDB по адресу localhost:27017.

Выгружены 9084 статьи для анализа при помощи MongoDB команд.

```
.find({date: {$gte: ISODate("2022-01-01T00:00:00.000Z")}})
.sort({date: 1})
```

Составлены ряд регулярных выражений и словарей для очистки шума в текстовых данных.

Полное удаление незначимой информации.

После подготовительной очистки необходимо сделать полную нормализацию данных:

- Перевод текста в нижний регистр;
- Полное удаление всех знаков пунктуации;
- Очистка всех цифровых символов и слов, которые их содержат;

- Очистка однобуквенных слов, так как такого рода слова несут минимум информации и могут восприниматься, как шум.
- Детальное удаление часто встречаемых слов при помощи словаря. Пример таких слов: «habr», «cnews», «dnews», «теги», «хабы» и другие слова;
- В конечно счёте применяется встроенный словарь стоп-слов для русского и английского языков, который удаляет все артикли, междометья, союзы, предлоги и прочее.

Векторизация информации.

После очистки шума и составления Bag of Words, можно применять частотный анализ текстовой информации $TF - IDF$. На основании результата алгоритма были выбраны 6 строк и составлена таблица.

Разметка данных

Следующим этапом после векторизации рассматривается построение моделей кластеризации и тематического моделирования. В ходе чего корпус текста был разбит на 3 кластера и 5 топиков. Было проанализировано разбиение на топики, выбраны ключевые слоа, характеризующие тематику документа. Аналогичным образом были проанализированны кластеры.

В итоге с применением метода главных компонент и узла визуализации была отображенна двумерная картинка, моделирующая разбиение документов на кластеры.

Разметка тональности текста

Чтобы произвести разметку текста необходимо составление словаря, содержащего ключевые слова, которые несут в себе либо отрицательную, либо положительную окраску.

В итоге был составлен словарь из 1524 слов несущих позитивную окраску и 3375 слова, которые относятся к негативной окраске.

В итоге разметка на основе составленных словарей с применением частотного анализа позволила получить коэффициент тональности текста $D_s = \frac{D_p - D_n}{n}$, где D_p - число положительных слов в документе, D_n - число отрицательных слов в документе, а n - общее число слов в документе и произвести конечную разметку текстов на тональные классы (POS/NEG).

Анализ полученных результатов

В качестве основных подходов анализа текста были выбраны следующие:

- Совмещение тонального анализа с кластерным и LDA, то есть рассматривается отдельно каждый кластер, выбирается наиболее тонально значимые статьи, как положительно, так и отрицательно и выбираются по 4 из каждого кластера (2 имеющих наибольшую положительную окраску, 2 имеющих наибольшую отрицательную окраску) и имеющих разные топики.
- Другой подход состоит в анализе наиболее приближенных и наиболее отдалённых от центра статей.

Это позволило выбрать 12 значимых статей на основе первого подхода анализа и 6 статей на основе второго подхода анализа.

Коротко резюмируя смысловое содержание выбранных статей, можно выделить основные тенденции:

- Переход на отечественные аппаратные средства и ПО.
- Внедрение систем для борьбы с кибератаками.

Таким образом, из результатов анализа можно сделать вывод, что в связи с уходом с Российского IT рынка большинства зарубежных компаний и оттоком специалистов, возникла необходимость наращивания производства отечественных аналогов технического и программного обеспечения. Также выявлена тенденция привлечения всё большего числа специалистов в IT индустрию, в частности в области по защите информации и созданию искусственного интеллекта.

Заключение. В магистерской работе были изучены основные принципы составления аналитических информационных систем от ETL процессов до полноценного анализа текстовой информации.

По итогу решены 2 задачи: разработка эффективного веб краулера; построение аналитической системы для анализа тенденций IT рынка.

В результате решения поставленных задач было собрано 19569 статей и проанализированно 9084 из всего списка путём отфильтровывания статей за 2022 год и упорядочивания их по дате. Этот подход позволил выявить наиболее значимые события, как в сфере Российского IT, так и некоторые сигналы из общего сектора.