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

Кафедра информатики и программирования

ПЛАТФОРМА ДЛЯ АВТОМАТИЧЕСКОГО УСТАНОВЛЕНИЯ УДК НАУЧНЫХ РАБОТ
АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 441 группы
направления 02.03.03 Математическое обеспечение и администрирование информационных систем
факультета компьютерных наук и информационных технологий
Лапушкина Дмитрия Алексеевича

Научный руководитель,
старший преподаватель кафедры ИиП _____ Е.Е. Лапшева

подпись, дата

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

подпись, дата

Саратов 2021

ВВЕДЕНИЕ

Актуальность темы. Каждому человеку, который занимается научной деятельностью, хотя бы раз в жизни приходилось оформлять научную работу в соответствии с жесткими требованиями. Это связано с множеством сложностей, одной из которых является проставление УДК. Из-за огромной иерархии классов в этой системе классификации порой бывает довольно трудно правильно и быстро определить итоговые «метки» в соответствии с данным стандартом для своей работы.

В нашем мире довольно давно существует множество алгоритмов, основанных на машинном обучении, которые проставляют метки для товаров в магазинах по принадлежности их к определенным классам, что стало для нас вполне обыденным. Поэтому создание платформы для автоматического определения УДК становится вполне выполнимой задачей, разрешение которой сможет упростить жизнь многим людям, занимающимся научной деятельностью.

На данный момент в открытом доступе не существует систем или алгоритмов, которые определяли бы УДК по тексту научной работы, а значит решение данной задачи по-прежнему актуально.

Цель бакалаврской работы – создать веб-приложение, принимающее на вход файл с текстом научной работы, обрабатывающее текст и приводящее его в вид пригодный для подачи на вход заранее обученным моделям машинного обучения, полученным в ходе дипломной работы Тимофеева Владислава, которые в свою очередь спрогнозируют её УДК. Помимо этого, так как для обучения моделей необходимы данные, а датасеты, пригодные для обучения модели, в открытом доступе в интернете не существуют, необходим парсер данных, который собирает данные для обучения моделей с сайтов, на которых в свободном доступе опубликованы научные работы с уже проставленным УДК. Вдобавок, для манипуляции классами УДК необходимо иметь всю иерархию, для получения которой необходим алгоритм, который сможет перенести эту иерархию в

систематизированный вид в коде, а затем сохранить для последующего использования.

Поставленная цель определила **следующие задачи**:

1. Провести анализ парсеров сайтов и способов их автоматизации, реализовать парсер для сбора необходимого количество данных, а также парсер для сайта, хранящего все классы УДК в соответствии с иерархией, для того, чтобы можно было удобно оперировать числовыми метками разделов.
2. Провести анализ алгоритмов анализа текста, выделяющих ключевые слова, выбрать подходящий и реализовать его.
3. Привести данные в пригодный для обучения моделей вид.
4. Создать web-приложение, связать его с обученными моделями машинного обучения.

Методологические основы алгоритмов извлечения ключевых слов представлены в работах Jurafsky D., H. Martin J., Bird, S., Klein E., Loper E., Lane H., Howard C., Max Hapke H., Mihalcea. R, Tarau P., Blei D., Ng A., Jordan M.

Практическая значимость бакалаврской работы.

Значимость данной работы заключается в возможности её последующего прикладного использования: платформа для распознавания УДК научной работы, полученная в результате данной бакалаврской работы может помочь людям, занимающимся научной деятельностью, сократить время при оформлении работ за счёт автоматического определения УДК.

Структура и объём работы. Бакалаврская работа состоит из введения, 5 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 81 страница, включая 23 рисунка, список использованных источников информации – 23 наименования.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Обработка естественного языка» посвящен описанию всех необходимых теоретических знаний для полного понимания работы практической части диплома, в которой происходит извлечение ключевых слов из текста. В целом, обработка естественного языка – область информатики и искусственного интеллекта, связанная с анализом текстовых данных на естественном языке, а именно с созданием и применением алгоритмов машинного обучения для анализа естественного языка. В данной работе рассматривается одна из задач анализа естественного языка – извлечение ключевых слов. Она подразумевает автоматическое определение в тексте списка слов или же словосочетаний, которые наилучшим образом описывают документ. Существует множество различных подходов для извлечения ключевых слов такие как лингвистические, основанные на графах и другие. Примерами методов извлечения ключевых слов, которые описаны в данной работе, являются алгоритм Text-rank, алгоритм, основанный на TF-IDF коэффициентах, а также алгоритм, использующий скрытое распределение Дирихле.

Text-rank – это алгоритм, основанный на Page-rank, который часто используется при извлечении ключевых слов и резюмировании текста.

Page-rank – это алгоритм, используемый для расчета веса веб-страниц. Можно рассматривать все веб-страницы как большой ориентированный граф, в котором узел – это веб-страница. Если веб-страница A имеет ссылку на веб-страницу B, ее можно представить как направленное ребро от A до B. Page-rank определяет числовую величину, которая характеризует «важность» веб-страницы по количеству ссылок на страницу, то есть чем их больше, тем она «важнее». Помимо количества ссылок на страницу немалую роль играют и «веса» этих ссылающихся страниц, то есть на «значимость» страницы A влияет вес ссылки, которая передается страницей B. Таким образом, Page-rank — это способ определения веса страницы за счет вычисления «важности» ссылок на неё. Основная идея, реализуемая моделью

ранжирования на основе графов, заключается в «голосовании» или «рекомендации». Когда одна вершина связывается с другой, она в основном голосует за эту другую вершину. Чем больше голосов отдано за вершину, тем выше ее важность. Кроме того, важность вершины, отдающей голос, определяет, насколько важен сам голос, и эта информация также учитывается моделью ранжирования. Следовательно, оценка, связанная с вершиной, определяется на основе голосов, поданных за нее, и оценки вершин, подавших эти голоса.

Второй алгоритм, использует для определения ключевых слов TF-IDF коэффициенты.

TF-IDF (Term Frequency & Inverse Document Frequency) – это числовая статистика, которая предназначена для отражения того, насколько важно слово для текста. Он часто используется для определения весового коэффициента при поиске информации, интеллектуальном анализе текста и пользовательском моделировании. Данный метод оценки «важности» термина в рамках текста является довольно удобным в использовании и простым в понимании. Принцип его работы заключается в том, что слово имеет большую значимость для документа тогда, когда оно встречается часто в каком-то определенном документе, при этом редко появляясь во всех остальных. Под документом можно понимать, как целый текст, так и отдельно взятое предложение, в зависимости от того, какого масштаба происходит анализ.

Третий рассматриваемый алгоритм использует для определения ключевых слов скрытое распределение Дирихле.

LDA (Latent Dirichlet allocation) или же скрытое распределение Дирихле – это генеративная статистическая модель, которая позволяет ненаблюдаемым группам объяснять наборы наблюдений, что позволяет понять, почему некоторые части данных схожи.

Наиболее часто LDA применяется для задач тематического моделирования. Тематическое моделирование — это задача использования

обучения без учителя для извлечения основных тем (представленных в виде набора слов), которые встречаются в коллекции документов. Таким образом, основные темы, извлеченные с помощью LDA, будут ключевыми словами.

Для того, чтобы применять к тексту LDA, необходимо преобразовать его в терм-документную матрицу.

Терм-документная матрица (term-document matrix) — это матрица, которая имеет размер N×W, где N — количество документов в корпусе, а W — количество уникальных слов, которые встречаются в тексте. В i-й строке, j-м столбце матрицы находится число, отображающее то сколько раз в i-м предложении, встретилось j-е слово.

Каждый из описанных алгоритмов использует токенизацию, стемминг, а также POS-теги.

Токенизация – один из основных и решающих этапов языковой обработки. Это обработка текста, которая подразумевает его разбиение на отдельно значимые единицы – токены. Очень часто токенизация используется при построении алгоритмов машинного перевода, а также для оптимизации различного рода поисковых систем.

Стемминг – это довольно грубый эвристический процесс, который производит удаление аффиксов (словообразовательных морфем) и оставляет только основу слова в соответствии с наборами правил, которые определяют, какие символы следует удалить или заменить. Зачастую стемминг используют вместо лемматизаторов специально для сокращения словаря.

В связи с тем, что многие алгоритмы, позволяющие определить ключевые слова в документе, подсчитывают частоту встречи данного слова в тексте, такие части речи как предлоги и союзы, конкретно в русском языке, могут внести определенный «шум»» очень частым употреблением. Цель этой части этапа обработки данных состоит в том, чтобы взять фрагмент текста и пометить каждый токен слова идентификатором POS (Parts-of-Speech – части речи). Данный этап имеет место быть только в случае токенов

уровня слов. Как правило, POS-теггер используется почти всеми библиотеками для NLP, что дает возможность включать и исключать типы речи из потенциально рассматриваемых алгоритмом ключевых слов.

Второй раздел «Универсальная десятичная классификация» посвящен описанию системы классификации информации, которая широко используется по всему миру для систематизации научных работ, литературных произведений и других видов документов. В нем рассказывается о структуре данной классификации, а также о специфике распределения тематик по разделам. Помимо этого, в разделе упоминаются системы, которые предоставляют возможность определения УДК «вручную», используя предоставляемые таблицы с иерархией данной классификации, по которой нужно самостоятельно «проходиться» в поисках соответствующего своей работе класса. Упоминание данных систем сделано для того, чтобы подчеркнуть актуальность данной работы, ведь системы, которая автоматически бы определяла УДК научной работы на просторах интернета не существует.

Третий раздел «Парсер и сбор информации» посвящен описанию необходимых терминов для понимания того, что такое парсинг в целом и какие бывают ограничения при его использовании. Парсинг — это сбор некоторой структурированной информации. Парсинг сайтов — это способ автоматического сбора больших объемов данных с веб-сайтов. Большая часть данных на сайтах не структурирована, находится в HTML формате. Благодаря алгоритму парсера, данные можно структурировать в соответствии с любым необходимым форматом, а также записать, например, в некоторую базу данных, чтобы их можно было затем использовать в своих целях. Парсер — программа, которая осуществляет сбор данных, структурирует их и сохраняет. Она переходит по заранее указанным ссылкам и собирает всю необходимую информацию. Несмотря на то, что автоматизация сбора данных позволяет сократить много времени, существует множество ограничений, которые могут значительно затруднить этот процесс. Самыми

распространенными видами ограничений при парсинге является блокировка по IP адресу, блокировка по User-agent, а также капча.

Четвертый раздел «Создание парсера для сбора данных и приведение их в необходимый вид для дальнейшего прогнозирования» посвящен реализации алгоритма парсера, описанию технологий, использованных при его создании, а также тому, как данные были систематизированы и обработаны.

К технологиям, использованным для создания парсера, относится фреймворк Selenium. По своей сути это набор инструментов для автоматизации веб-браузера, в котором используются лучшие доступные методы для удаленного управления экземплярами браузера и имитации взаимодействия пользователя с браузером. Это позволяет разработчикам моделировать общие действия, выполняемые пользователями: ввод текста в поля, выбор раскрывающихся значений и флажков, а также нажатие на ссылки в документах.

Для того чтобы собрать как можно больше данных было решено сделать парсер сразу для нескольких сайтов, публикующих научные работы в открытом доступе: «Научная электронная библиотека eLIBRARY.RU» (https://elibrary.ru), «ПОЛИТЕХ. Электронная библиотека» (https://elib.spbstu.ru), а также и электронная библиотека СНИГУ им. Чернышевского «Зональная научная библиотека им. В.А.Артисевич» (http://library.sgu.ru). При написании был использован Page-Object подход, что подразумевает описания действий для каждого типа веб-страницы в отдельном классе.

В итоге был собран датасет, состоящий 160 тысяч образцов данных, каждый из которых имеет такие параметры как название научной работы, ключевые слова, размеченные сайтом, а также значение УДК в числовом формате. Из-за огромного количества разделов в УДК в полученном датасете было множество классов, в которых число экземпляров не превышало 5 образцов, чего точно недостаточно для обучения. В связи с этим данные

были обработаны следующим образом: метки УДК у образцов были проставлены на класс выше по иерархии, в случае если образцов для данного класса было собрано менее 15. В итоговом варианте данных образцы имеют тот же вид, что и в изначальном, отличается лишь количество классов, которое теперь равно 560. Далее в данном разделе идет описание программного кода, выполняющего парсинг.

Пятый раздел «Веб-приложение, реализующее обработку текстового документа» посвящен реализации трех алгоритмов для извлечения ключевых слов из текста научной работы. Помимо этого в нем описывается реализация веб-приложения, а также технологии, использованные при создании. К таким технологиям относятся фреймворк Spring и шаблонизатор Freemarker. Spring – довольно простой в использовании мощный фреймворк для разработки приложений. Это хорошо проработанный инструмент, который поддерживает несколько веб-приложений, использующих Java в качестве языка программирования. Freemarker – это компилирующий обработчик шаблонов, написанный на языке Java. Он позволяет описывать веб-интерфейс, используя подход «Model-view-controller», что дает возможность использовать внутри html кода параметры модели, заданные в контроллере. Данный шаблонизатор используется для того, чтобы выносить повторяющиеся участки кода веб-интерфейса в отдельные файлы для возможности их повторного использования. Freemarker способен преобразовывать специальные выражения, такие, как, итерации по коллекции, и динамически заполнять документ данными, которые соответствуют этим выражениям.

Для данной работы был создан отдельный контроллер UploadApiController.java, осуществляющий логику обработки запросов к данному приложению.

Для того чтобы запустить сторонний скрипт из java-кода, в данном случае из контроллера, необходимо воспользоваться чем-то, что умеет манипулировать потоками: создавать или завершать поток, а также

перехватывать информацию из него. В этой работе был использован объект класса ProcessBuilder. Для использования необходимо создать экземпляр данного класса и передать ему путь к исполняемому скрипту, а также параметры, с которыми необходимо его запустить. Далее в этом разделе идет описание кода скрипта для извлечения ключевых слов из текста, а также описание того, как происходит взаимодействие между веб-приложением, написанном на языке Java, и скриптом для извлечения ключевых слов, написанном на языке Python.

ЗАКЛЮЧЕНИЕ

Целью бакалаврской работы было создание web-платформы, которая смогла бы помочь людям, занимающимся научной деятельностью, с установлением УДК их научной работы, опираясь на текст этой работы. В рамках данной работы был проведен анализ литературы (9 русскоязычных источников и 14 англоязычных) по темам: парсинг, NLP, извлечение ключевых слов. На основе полученной базы знаний были созданы несколько парсеров под определенные сайты, публикующие научные работы в открытом доступе, а также парсер для иерархии УДК. Помимо этого, было создано web-приложение, извлекающее ключевые слова из загружаемого текста и передающего эти ключевые слова для прогнозирования УДК на вход моделям машинного обучения, которые описаны Тимофеевым Владиславом в его дипломной работе. В будущем планируется размещение данного приложения на постоянном сервере для возможности доступа к приложению через интернет. Деятельность, проведенная в рамках дипломной работы, имеет развитие в дальнейшем: покупка API на сайтах, где научные работы находятся в закрытом доступе помогло бы увеличить базу данных, что в свою очередь может помочь усовершенствовать данную платформу и сделать её более продуктивной.

Основные источники информации:

1. Jurafsky, D.Speech and Language Processing / Jurafsky D., H. Martin J., Alan Apt, 2000. — С. 221-234.

2. Bird, S. Natural Language Processing with Python / Bird, S., Klein E., Loper E. // O'REILLY, 2009. — С. 55-69.

3. Lane, H. Natural Language Processing in Action Understanding, analyzing, and generating text with Python / Lane H., Howard C., Max Hapke H. // Manning Publications Co., 2019. — С. 120-150.

4. Mihalcea, R TextRank: Bringing Order into Texts / Mihalcea. R, Tarau P. // University of North Texas Press, 2004. — С. 2-5.

5. Blei, D. Latent Dirichlet Allocation / Blei D., Ng A., Jordan M. // Journal of Machine Learning Research 3, 2003. — 251 с.

6. Парсер на Selenium WD [Электронный ресурс]: Хабр — URL: https://habr.com/ru/post/186496/ (дата обращения: 21.11.2020) — Загл. с экрана. — Яз. рус.

7. Walls, C. Spring in action / Walls C. // Manning Publications Co., 2011. — С. 327-368.

8. FreeMarker Java Template Engine [Электронный ресурс]. URL: Freemarker project — https://freemarker.apache.org/ (Дата обращения 20.10.2020). — Загл. с экрана. — Яз. англ.