ОГЛАВЛЕНИЕ

1. Введение
2. Описание предметной области
3. Обзор методов кластеризации  
   1. Общий подход к кластеризации объектов  
   2. Виды метрик в кластерном анализе
4. Проектирование системы  
   1. Анализ предметной области  
   2. Выявление определяющего фактора ключевых слов  
   3. Определение требований к системе  
   4. Применение метода Data Mining
5. Реализация Системы  
   1. Структура БД и хранилищ данных  
   2. Реализация алгоритма  
   3. Пользовательский интерфейс  
   4. Описание работы системы
6. Тестирование  
   1. Обучение системы  
   2. Анализ результатов  
   3. Корректировка алгоритма  
   4. Тестирование с произвольными данными
7. Выводы
8. Литература
9. Приложение

ВВЕДЕНИЕ

На сегодняшний день уже очевидно, что в будущем наиболее распространенный вид работы будет являться сочетанием человеческих и машинных ресурсов. Уже сейчас такой тандем активно практикуется в различных сферах жизнедеятельности человека. Этот симбиоз позволит значительно увеличить качество и скорость работы за счет использования вычислительной мощности техники и уникальных качеств человеческого организма.

При использовании такого подхода машина (компьютер) должна выполнять роль механизма анализа накопленных (собранных) данных и инструментом прогнозирования, человек же — корректирующей системой. Такой метод работы с информацией сможет максимально рационально использовать все лучшие качества каждого инструмента.

При выполнении дипломной работы, была построена одна из таких систем с учетом их особенности и нюансов, связанных с сферой образования, в частности с распределением нагрузки кафедры. В ходе исследования и сравнительной характеристики будут определены методы и пути решения данной проблемы

Путем анализа научных трудов можно выявить новую информацию о квалификации преподавателя в различных отраслях знаний и упростить задачу работников кафедры предоставляя им более подробную информацию.

Таким образом можно получить информацию, которая позволит намного точнее рассказать об опыте субъекта в данной отрасли знаний, тем самым позволив работнику принять более взвешенное решение.

Целью исследования является оптимизация распределения нагрузки кафедры на основе анализа публикаций сотрудников кафедр, их персональной информации и кафедральной документации. Путем анализа трудов сотрудников сферы образования можно выявить новую информацию об их знания в различных сферах, тем самым увеличить качество распределения нагрузки.

Таким образом объект исследования будет являться процесс сбора, хранения и анализа данных принимающих участие в формировании нагрузки кафедры, а предметом – подсистема сбора и анализа публикаций сотрудников кафедр системы управления кафедрой.

Для достижения поставленной цели необходимо решить следующие задачи:

1. Изучить и проанализировать процессы протекающие в сфере образования, в рамках задачи оптимизации распределения нагрузки кафедры
2. На основании данных полученных при решении первого пункта вывести требования к системе, ПО, и внедряемым алгоритмам
3. Проанализировать алгоритмы Data Mining, которые уже активно используются для решения подобных задач. В случае, если ни один алгоритм не сможет решить проблему в полной мере предлагается рассмотреть применение смешанных алгоритмов либо модификацию существующих.
4. Реализовать и протестировать выбранный (разработанный) алгоритм.
5. На основании полученных данных после тестирования сделать выводы позволяющие оптимизировать алгоритм и внедрить его в систему распределения нагрузки кафедры модульно.

ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ

Сфера образования является одной из наиболее важных сфер жизнедеятельности человека. Это обусловлено тем, что ВУЗы готовят специалистов, которые будут в будущем заниматься развитием человечества. Поэтому крайне важно поддерживать качество образования на высоком уровне.

Однако, с увеличением количества новых предметов в вузах можно увидеть, что сотрудникам данной сферы, с каждым годом, приходится оперировать все большими объемами данных, что неизбежно приводит к увеличению ошибок.

Задача распределения нагрузки является одной из наиболее трудоемких и важных в учебном процессе и, соответственно, она не может быть абсолютно защищена от человеческого фактора.

Также, стоит отметить, что за долгие годы, было накоплено множество печатной и электронной неструктурированной информации с которой крайне сложно работать.

Таким образом, если структурировать накопленные знания и упорядочить всю информацию, можно будет повысить осведомленность сотрудников об квалификации преподавателей и тем самым увеличить показатели качества обучения. Этого можно достичь если предоставлять работникам информацию в привычной для человека форме и впоследствии облегчая задачу распределения нагрузки.

Ранее уже была разработана система позволяющая выполнять операции в online режиме путем взаимодействия с системой менеджмента [1]. Однако, система только повторяла ручной процесс распределения нагрузки.

На сегодняшний день становятся все более актуальным использование различных инструментов интеллектуального анализа для оптимизации человеческого труда. Такой подход значительно увеличивает качество работы. Анализируя предметную область, выявляя зависимости и обучаю машину работать в паре с человек такой подход получил широкое распространение в сферах медицины, рекламы и бизнеса. Сбор и анализ данным помагает выявить общие тенденции и выявлять зависимости еще на ранних этам процессов, что значительно ускоряет работу и ускоряет принятие решений. Однако, стоит отметить, что данный подход, в связи с сложностью исполнения, все еще является весьма трудоемким и дорогим в разработке.

В скором будущем человеку придется обрабатывать в миллионы раз больше информации чем мы уже накопили сейчас. Объем информации возрастает ежегодно на 30% что около 2,5∙108 байт на человека [2]. На основании статистики, собранной за долгое время, уже давно был выведен и освещен закон роста информации в работах Дерека Прайса и описан как экспоненциальная функция (1).

Очевидно, что при таком росте информации необходимы инструменты ее обработки, иначе ее рациональное использование становится невозможным.

Существует множество методов анализа данных, а вся отрасль, занимающаяся анализом данных, называется Data Science [3]. Этот раздел знаний активно занимается применением различных методов и подходов к обработке и анализу данных. Сочетая использование вычислительных мощностей компьютеров так и законы линейной алгебры и статистики становиться возможным уже сейчас производить анализ терабайтов данных.

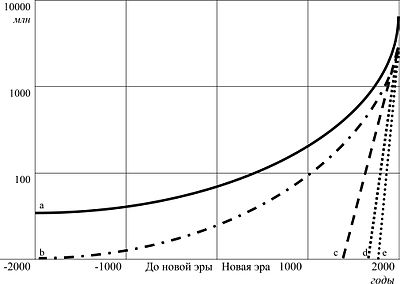


Рисунок 1. График увеличения накопленной информации относительно времени.

Одним из направлений Data Science является Data Mining. Объединяя в себе совокупность методов для решения различных аналитических задач, этот подход уже активно используется для интеллектуального анализа различными крупными компаниями. Например Google для индексации страниц в своём поисковике используют метод Page Rank, а Amazon используют наборы методов для рекламы своих продуктов [4].

В целом, методы Data Mining делятся на два основных типа по их применению: прогнозирующие и описательные [5].

Каждая группа состоит из методов которые обладают рядом определенных свойств. Например, к прогнозирующим методам относятся методы Дерева Решения, Нейронные Сети, Методы Регрессии, а к описательным методы кластеризации и классификации.

Прогнозирующие методы предназначены для анализа возможностей в будущем на основе накопленных данных. Такие методы можно использовать, например, для предсказания успеваемости студента на основе его предыдущих успех.

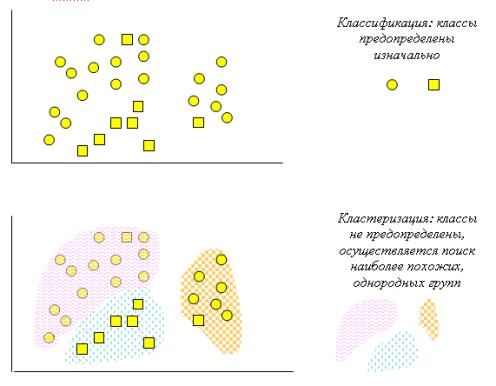
Описательные методы используются для разделения объектов на группы, что позволяет находить закономерности между объектами и получать новую информацию.

Так как, входными параметрами для оптимизации распределения нагрузки являются научные тексты, можно, проанализировав их свойства, объединить их в группы схожести, тем самым получив новую информацию о преподавателях.

Методы классификации применяются для отраслей в которых изначально заданы точные модели данных, относительно которых будет работать алгоритм (Рис. 1). Благодаря этому, алгоритм работает весьма быстро но имеет недостаток связанный с жесткостью задания условий.

Кластеризационные методы используются в отраслях где заведомо неизвестны особенности каждой новой группы (Рис. 1) Алгоритм распределяет данные по группам схожести основываясь на выбранной метрики. Данные методы требуют больше временных затрат, но являются более гибкими и расширяемыми.

Также, для создания внутренней иерархии кластеров используется метод иерархической кластеризации, благодаря которому можно получить данные связанные между собой родительно-наследственными связями, что позволит сделать алгоритм точнее.

Рис. 1. Иллюстрация различий выборки методов кластеризации и методов классификации.

В каждом из методов Data Mining можно использовать алгоритмы самообучения, которые позволят определять новые данные с высокой точность. С внедрением такого алгоритмы в метод иерархической кластеризации можно, в ходе его работы, получать новые кластеры и делать разбиения точнее с каждой новой итерацией.

Для реализации подобных алгоритмов обычно используются микросервисы благодаря которым, в зависимости от необходимости, можно, в любую единицу времени, увеличивать или уменьшать вычислительную мощность, что позволяет сделать подсистему более гибкой и надежной.

Произведя анализ области знаний было выявлено, что сотрудники занимающиеся нагрузкой кафедры не всегда и не точно знают о квалификации преподавателей в различных сферах, что затрудняет распределение нагрузки и тратит время. Однако, каждый научный деятель постоянно пишет множество статей и трудов, анализ которых мог бы значительно облегчить задачу распределяющих нагрузку.

В рамках поставленной задачи можно использовать методы кластеризации для распределения научной литературы преподавателей по различным кластерам, тем самым увеличивая осведомленность работников, занимающихся нагрузкой, о достижениях их коллег.

Отображая полученную информацию в удобной и привычной форме, можно еще на этапе распределения нагрузки увеличивать качество обучения, так как на каждую дисциплину будет автоматически рекомендоваться преподаватель с наиболее высокими оценками в выбранной дисциплине.

ОБЗОР МЕТОДОВ КЛАСТЕРИЗАЦИИ

Общий подход к кластеризации объектов

Кластеризация (или кластерный анализ) — это задача разбиения множества объектов на группы, называемые кластерами[6]. Внутри каждой группы должны оказаться объекты с близкими параметрами схожести, а объекты разных группы должны быть как можно более отличны. Главное отличие кластеризации от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных.
3. Вычисление значений меры сходства между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата.

Одним из наиболее популярных и распространенных методов является метод K-средних (k-медиан), За счет простоты реализации и скорости работы он уже давно проявил себя в задачах распознавания объектов. Однако, этот метод обладает одним главным недостатком – изначально заданным количеством кластеров. В некоторых ситуациях можно изначально, достаточно точно, предугадать максимальное количество возможных кластеров, но такой подход накладывает значительные ограничения на развитие системы в будущем.

Близким родственником этого k-медиан является алгоритм N-средних (n-means). Он предполагает, что количество кластеров заведомо и позволяет расширять их количество по мере накопления данных. Однако одним из существенных недостатков является ресурсозатратность, так как, необходимо каждый раз находить новые центроиды и строить новые кластеры.

Так как в рамках решаемой задачи неизвестно количество кластеров и их особенности было решено использовать именно метод N-средних, это позволит с каждым разом находить все больше полезной информации о знаниях преподавателей.

Стоит отметить, что методы, также, делятся на самообучаемые и на обучаемые с учителем. За счет того, что объемы данных ожидаются незначительными методы с самообучением могут быть малоэффективны и весьма трудозатратны. Поэтому, оптимально будет использовать возможность явно корректировать алгоритм.

Метрик в кластерном анализе

Для определения объектов в методах кластеризации используются специальные метрики, или функции расстояния. Эти расстояния могут определятся в одномерном или многомерном пространстве и являются описательной характеристикой схожести объектов.

В основном, используется 3 наиболее популярных видов измерений:

1. Евклидово расстояние (Рисунок 1)
2. Расстояние городских кварталов (манхэттенское расстояние) (Рисунок 2)
3. Процент несогласия (Рисунок 3)

Первый вариант является наиболее популярным и активно используется для кластеризации изображений. Использование его оправданно в тех случаях, когда признаки измеряются в одинаково и имеют схожий вес [7].

Рисунок 1. Формула евклидового расстояния

Использование расстояния городских кварталов зачастую связанно с уменьшением влияния нежелательных выбросов. За счет отсутствия степени данная формула более устойчива к отдельным большим разностям, в остальных случаях приводит к схожему результату с евклидовым расстоянием.

Рисунок 2. Формула манхэттенского расстояния

Так как не все признаки можно измерить числовыми значениями для таких случаев используется формула процента несогласия. Она эффективно работает в анализа текста и ключевых слов, а также при анализе анкет или другой схожей информации [8].

Рисунок 3. Формула процента несогласияP~=~VALUE~delim{|}{~A_{i}~<>~~B_{i}}{|}

Входными параметрами для подсистемы оптимизации распределения нагрузки кафедры входными параметрами являются именно тексты. И путем анализа ключевых слов и нахождения различив в словах, в разных текстах, можно получить процент отличия каждого текста от другого, что позволит выявить центроиды и построить кластеры.

ПРОЕКТИРОВАНИЕ СИСТЕМЫ

Анализ предметной области

В ходе анализа предметной области было выявлено, что одной из наиболее сложных задач является построение соотношений межды предметами и преподавателями. Часто возникают с трудностями и выявлениями знаний преподавателя в той или иной дисциплине. Таким образом было решено построить систему, которая будет давать советы о квалификации сотрудника высшего учебного заведения на основании его научных работ.

Такой подход позволит не только облегчить работы над распределением нагрузки, а также сможет структурировать накопленные научные работы и разбить их по сферам знаний.

Для анализа таких текстов нужно брать во внимание, что научная литература значительно отличается от других и наиболее важным для понимания смысла текста являются ключевые слова. Находя такие слова в тексте и сравнивая их наличие с другими текстами можно будет заметить значительные различия между текстами, например, о теории вероятностей и программированием.

Также, анализ показал, что объемы данных будут незначительными, что позволит применить метод с неизвестным числом кластеров. Однако, потребуется достаточно внушительная база обучения, которая сможет в достаточной мере покрыть все слова-исключения.

Таким образом, разработанная система будет выполнять несколько функций :

1. Сбор и хранение данных
2. Интеллектуальный анализ
3. Рекомендации на основе произведенного анализа.

Выявление определяющего фактора ключевых слов

Как было изложено ранее, одним из наиболее важных этапов в анализе текстов является этап поиска ключевых слов. Для этого необходимо определить, что будет являться ключевым словом, а что – нет.

Ключевое слово— слово в тексте, способное в совокупности с другими ключевыми словами дать высокоуровневое описание содержания текстового документа, позволяющее выявить его тематику [9].

Таким образом, можно определить, что в основном ключевыми словами будут являться существительные и термины. Однако, стоит принимать во внимание и другие части речи, так как они тоже могут нести в себе смысл [10]. Также, стоить помнить о различиях в терминологии в разных сферах и учитывать особенности каждой для улучшения показаний алгоритма. Так, например, большая часть ключевых слов в отрасли программирования будет описана латиницей.

Зачастую, для распознавания частей речи используют 2 подхода : используют базу-словарь или нейронную сеть. В первом случае это влечет за собой большие объемы данных и сложность работы с новыми, еще не определенными словами. Второй же является крайне трудозатратным. Поэтому в работе будет использоваться смешанный алгоритм, который будет поддаваться ручной корректировке.

Еще одним затруднением могут быть коллизии в словах. Во многих языках, одним и тем же словом-термином могут называть разные вещи из разных отраслей. Поэтому необходимо хранить связи между ключевыми словами и уже определенными сферами их использования.

В итоге, очевидно, что необходимо хранить:

1. Данные о ключевых словах
2. Данные о сферах применения ключевых слов
3. Соотношение ключевых слов к тексту

Эта информация позволит в достаточной степени получать представление

Определение требований к системе

В ходе анализа предметной области было выявлено, что система должна заниматься сбором, хранением и анализом данных. Ввиду этого, было решено использовать облачное хранилище для размещения загружаемых файлов, а для установления связей между ключевыми словами и отраслями знаний добавить новые таблицы уже в существующую базу данных.

Так как процесс анализа является весьма ресурсо и времязатратным, было решено создать отдельный микросервис и разместить алгоритм на нем. Такое решение позволит динамически масштабировать процессы и выполнять операции параллельно, а также снизить нагрузку на основной сервер.

Таким образом система должна удовлетворять следующим требованиям:

1. Надежность
2. Доступность
3. Масштабируемость
4. Совместимость с остальным функционалом системы
5. Удобство в использовании

Для удобства использования, было решено внедрить в интерфейс раздела распределения нагрузки, в момент создания новой записи информацию о преподавателях и сортировать их в зависимости от выбранного предмета.

Также, был предусмотрен интерфейс для корректировки системы анализа, показывающий количество кластеров, их содержимое и ключевые слова, которые были использованы для анализа текстов.

Таким образом, система состоит из следующих модулей:

1. Модуль сбора данных, расположенный на основном сервере
2. Модуль хранения данных, являющийся базой данных PostgreSQL и облачных хранилищем.
3. Модуль анализа данных и создания новых ключевых слов, расположенный на отдельном виртуальном сервере
4. Интерфейс в разделе распределения нагрузки
5. Интерфейс корректировки алгоритма анализа данных

Одним из важных факторов для системы является надежность и доступность. Поэтому было решено использовать сервис AWS S3 как облачное хранилище для научных трудов. Данный сервис обеспечивает сохранность данных, безотказность. Одним из наиболее важных факторов является его совместимость с уже существующим окружением системы.

Так как система использует сервисы heroku было решено использовать все модули имеющие совместимость с данным сервисом.

В дальнейшем, для обеспечения более гибких настроек, система будет перенесена полностью на AWS. Благодаря тому, что heroku базирует все свои сервисы на Amazon это можно будет сделать без потери данных[11].

Применение метода Data Mining

Для решения поставленной задачи за основу был выбран метод кластеризации N-means, так как невозможно определить количество кластеров заранее. Также, в качестве меры расстояний между объектами был выбран процент несогласия между ключевыми словами входящими в каждый текст.

Весь алгоритм будет делиться на следующие основные фазы:

1. Извлечение файлов
2. Поиск ключевых слов и исключений
3. Расчет расстояний между объектами в выборке
4. Поиск центроидов
5. Распределение оставшихся объектов по кластерам, основываясь на максимальном проценте несогласия для каждого центроида

Таким образом, первым главным этапом является распознавания слов. Для определения ключевых слов было решено использовать несколько функций. Первая функция определяет ключевые слова с помощью библиотеки keyword-extractor. Она основана на основных особенностях построения речи в разных языках. Вторая функция отсеивает ключевые слова окончания которых не относятся к существительным и проверяет на стандартные исключения. Последняя — проверяет наличие слов в базе, убирает из выборки ключевые слова которые были вручную исключены и создает новые записи в базе при необходимости (Рисунок 1).

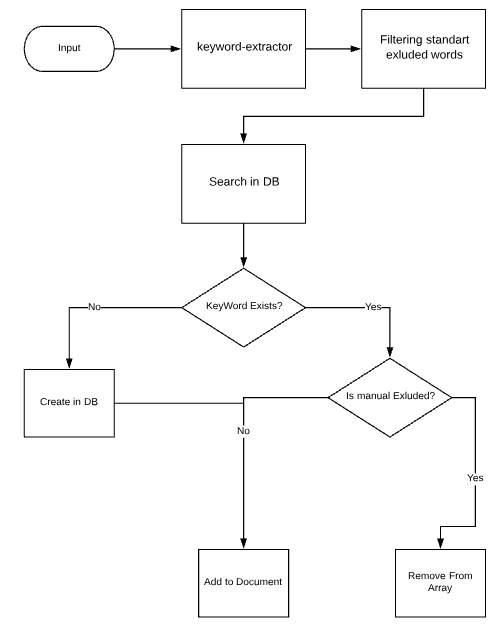
Расчет расстояний будет проходить по следующей схеме: так как необходимо построить отношение всех объектов ко всем, очевидно, что на выходе получится симметричная матрица с главной диагональю, состоящей из нулей. Это обусловлено тем, что расстояние (разница) между двумя одинаковыми элементами нулевая. Таким образом нужно вычислить только значения одной “половинки” матрицы. В общем виде сложность алгоритма будет равна что значительно меньше нежели обход по каждому элементу. Более того, данная операция не требует последовательности действий и является изолированной, что позволяет ее распараллелить на несколько потоков. 

Рисунок 1. Схема поиска ключевых слов.

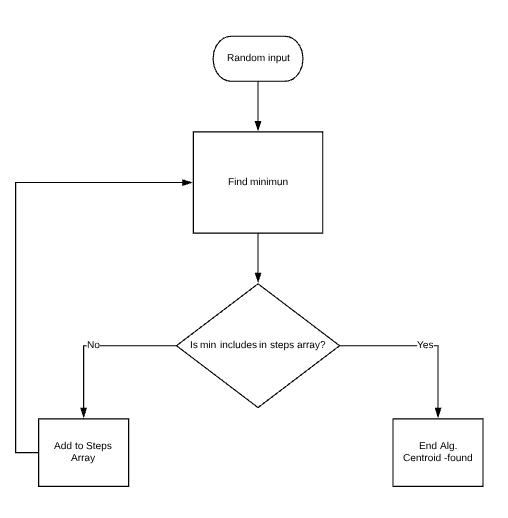
После получения матрицы значений необходимо найти центроиды. Центроидом будет являться точкой сгущения объектов, то есть точка с минимальным расстоянием от всех близлежащих. Поиск центроида будет находится похожим образом как в методе K-means, путем выбора произвольной точки и нахождения минимального расстояния между точками до того момента, пока не появится зацикливание (Рисунок 2).

Рисунок 2. Схема поиска ценроидов кластера

После нахождения центроида остается только отнести объекты к найденному кластеру в соответствии с максимальным несогласием.

Данный процесс продолжается до тех пор, пока не будут распределены все файлы из выборки.

РЕАЛИЗАЦИЯ СИСТЕМЫ

Работает - не трогай.

Для реализации системы был создан отдельный репозиторий и была настроена автоматическая заливка кода на работающий инстанс. Таким образом сохраняется вся история изменений кода и поддерживается целостность системы. Также это значительно облегчает работу с серверным приложением.

Как уже говорилось, в качестве виртуального сервера будет использоваться PaaS Heroku, который позволяет удобно, быстро и недорого настраивать все необходимые компоненты. Таким образом был настроен PostgreSQL и AWS S3 для хранения данных работы алгоритма.

Структура БД и хранилищ данных

Для реализации алгоритма было создано несколько миграций для базы данных основной системы (Приложение А). Так как в системе используется реляционная база данных PostgreSQL это позволяет сохранить жесткую связку между всеми старыми и новыми сущностями.

В ходе исследования было определено что необходимо создать таблицы для keywords (ключевых слов), knowledges (отраслей знаний), clusters (таблица для хранения данных о созданных кластерах) и таблица articles (научные работы и ссылки на файлы в их хранилище). Также, необходимо создать промежуточный таблицы, которые будут осуществлять связь многие-ко-многим с таблицами keywords-knowledges, knowledges -lectors, keywords-articles (Рисунок 1). Путем создания этих таблиц было жестко связано каждый элемент рабочей цепочки.

Рисунок 1. Тут будет Схема БД (а может и в Приложении)

В облачно файловом хранилище был создан bucket в который складываются все новые документы. Объем файлового хранилища пока составляет 5 Гб, но стоит отметить, что при необходимости его можно легко увеличить. Для создания хранилища было необходимо выполнить одну команду [12] :

heroku addons:create bucketeer:standard --bucket-name

Далее все необходимые параметры доступа можно получить из консоли Heroku (Рисунок 1).

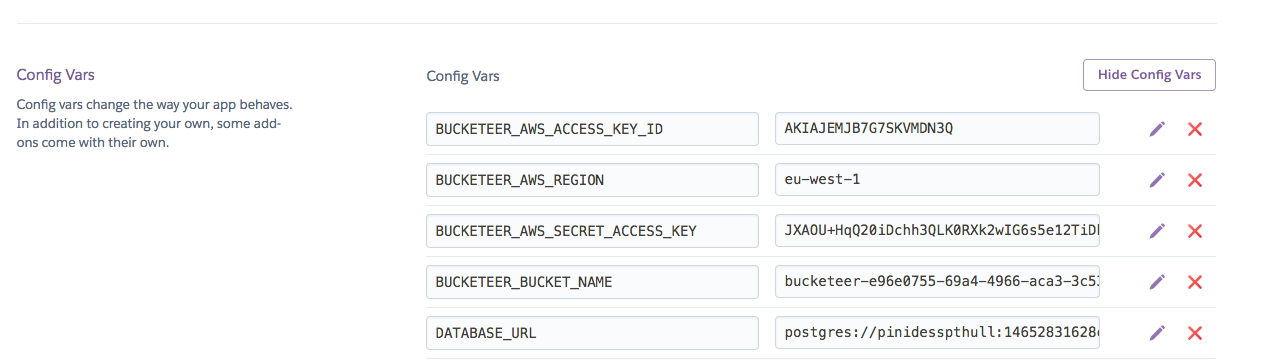


Рисунок 1. Параметры доступа к S3 Bucket из консоли Heroku

Далее было необходимо создать API которое позволяло бы заливать новые файлы в созданное хранилище. Для этого был написан новый путь в стиле REST для заливки файла от имени преподавателя :

POST /api/v1/admin/lectors/:lector\_id/files/:file\_type

Такой подход к созданию API позволяет сохранять очевидные пути доступов к объекту и поддерживать версионность. Таким образом, если в дальнейшем будет необходимо создать мобильное приложение и оно будет строго привязано к этому API его можно будет заменить на новое для новых версий приложений, а старое оставить для поддержки :

POST /api/v2/admin/lectors/:lector\_id/files/:file\_type

После написания API и настройки всех middlewares для распарсивания входного потока данных, был реализован контроллер, содержащий логику сохранения файла в БД и на S3 (Приложение Б). необходимо отметить, что все операции выполняются транзакционно, что позволяет в случае ошибок не сохранять неактуальные данные в разных таблицах и защитить код от поломки.

Таким образом был написан алгоритм заливки файла в облачное хранилище и создания всех необходимых записей в БД.

Реализация алгоритма

После создания записей в БД и существовании залитых файлов на S3 необходимо их оттуда считывать/скачивать для дальнейшей обработки.

Для решения этой задачи был написан сервис основанный на AWS SDK который позволяет легко взаимодействовать с API Amazon:

class S3Service {

constructor() {

this.s3 = new AWS.S3({

accessKeyId: config.get('AWS.S3.accessKeyId'),

secretAccessKey: config.get('AWS.S3.secretAccessKey'),

params: { Bucket: config.get('AWS.S3.bucket') }

});

}

/\*\*

\* Params Object like for amazon S3 method

\* @param {\*} params

\*/

put(params) {

return Promise.fromCallback(cb => this.s3.putObject(params, cb))

}

/\*\*

\* Params Object like for amazon S3 method

\* @param {\*} params

\*/

get(params) {

return Promise.fromCallback(cb => this.s3.getObject(params, cb))

}

getFile(params) {

return Promise

.fromCallback(cb => this.s3

.getSignedUrl('getObject', params, cb))

}

Именно этот сервис является оболочкой (оберткой) на AWS SDK и позволяет работать с ним в инкапсулированном виде. такой подход обеспечивает единую точку контроля над вариантами функций для чтения файлов.

После того как файл считан и сохранен его необходимо считать для распарсивания. Для этого была написана функция, которая считывает файл, разбивает его на слова, производит поиск ключевых слов и возвращает объекты ORM Bookshelf.js, ссылающиеся на записи в БД, для более удобной работы с ними [13].

ВЫВОДЫ

В ходе дипломной работы был реализован и внедрен алгоритм Data Mining, который позволил создавать рекомендации относительно соотношения преподаватель-предмет. Такой подход позволяет всегда информировать сотрудников кафедры о новых достижениях преподавателей в разных дисциплинах тем самым повышая качество распределения нагрузки.

Данный алгоритм не является совершенным, но даже при скромных объемах он показывает хорошие результаты. Его можно улучшить заменив существующие функции для поиска ключевых слов на нейронную сеть, что позволит куда точнее распознавать наиболее важные для расчетов слова. Также увеличивая обучающую выборку алгоритм тоже может быть доработан.

Построенный алгоритм может широко использоваться не только в сфере образования, а еще в сфере рекламы и маркетинга, ориентируясь на тематику текста и показывая его на сайтах со схожим контентом.

В дальнейшем планируется перенести всю систему на AWS что позволит взаимодействовать с виртуальными машинами на низком уровне, что в свою очередь даст возможность устанавливать и создавать более оптимизировано ПО. Также, планируется разработка мобильного приложения, которое поможет и преподавателям и студентам всегда находиться в курсе событий.

ЛИТЕРАТУРА

1. Тезисы Царюк А.О. 2016 стр. какая-то

# Lyman P., Varian H.R. [How much information](https://chnm.gmu.edu/digitalhistory/links/pdf/preserving/8_5a.pdf). Release of the University of California. Oct.27, 2003.

# Data Science from Scratch [Joel Grus](http://www.oreilly.com/pub/au/6494) Publisher: [O'Reilly Media](https://www.safaribooksonline.com/library/publisher/oreilly-media-inc/?utm_medium=referral&utm_campaign=publisher&utm_source=oreilly&utm_content=catalog&utm_content=catalog) Release Date: April 2015 Pages: 330 / 9 старница

1. Matthew Richardson, Amit Prakash, Eric Brill. [Beyond PageRank: Machine Learning for Static Ranking](http://www2006.org/programme/files/xhtml/3101/p3101-Richardson.html). — 2006.
2. Электронный ресурс. <http://intellect-tver.ru/?p=165>
3. Мандель И. Д. Кластерный анализ. — М.: Финансы и Статистика, 1988.
4. Электронный ресурс <https://habr.com/post/101338/>
5. Электронный ресурс <http://www.aiportal.ru/articles/autoclassification/measure-distance.html>
6. 1. Ванюшкин А.С. Методы и алгоритмы извлечения ключевых слов / А.С. Ванюшкин, Л.А. Гращенко // Новые информационные технологии в автоматизированных системах. – 2016. – № 19. – С. 85-93.
7. Тезисы о ключевых словах 2018
8. Электронный ресурс <https://devcenter.heroku.com/categories/heroku-architecture>
9. Электронный ресурс <https://devcenter.heroku.com/articles/bucketeer>
10. Электронный ресурс <http://bookshelfjs.org/>

ПРИЛОЖЕНИЕ А

Код миграций для начальных кластеров и связей

"use strict";

exports.up = function (knex) {

return knex.schema

.createTable("clusters", function (table) {

table.increments("id").primary();

table

.string("name");

table

.string("color");

table

.string("description");

table

.double("difference")

.notNullable();

table

.bigInteger("workspace\_id")

.notNullable()

.references("workspaces.id")

.onDelete("CASCADE");

table

.timestamp("created\_at")

.defaultTo(knex.fn.now())

.notNullable();

table

.timestamp("updated\_at")

.defaultTo(knex.fn.now())

.notNullable();

});

};

exports.down = function (knex) {

return knex.schema

.dropTable("clusters")};

Код миграций для начальных для таблицы Articles

'use strict';

exports.up = function (knex) {

return knex.schema

.createTable('articles', function (table) {

table

.increments('id')

.primary();

table

.string('mime\_type');

table

.string('name')

.notNullable();

table

.string('s3\_key');

table

.bigInteger("lector\_id")

.notNullable()

.references("lectors.id")

.onDelete("CASCADE");

table

.bigInteger("cluster\_id")

.references("clusters.id");

table

.timestamp('created\_at')

.defaultTo(knex.fn.now())

.notNullable();

table

.timestamp('updated\_at')

.defaultTo(knex.fn.now())

.notNullable();

table

.unique(['name', 'mime\_type', 'lector\_id']);

})

};

exports.down = function (knex) {

return knex.schema

.dropTable('articles')

Код миграций для для добавления центроида к кластеру

const Promise = require('bluebird');

exports.up = function (knex, Promise) {

//Users Table

return knex.schema.table('clusters', function (table) {

table

.bigInteger("centroid")

.notNullable()

.references("articles.id");

});

};

exports.down = function (knex, Promise) {

return knex.schema.table('clusters', function (table) {

table.dropColumn('centroid');

})

};

exports.config = {

transaction: true

};

};

ПРИЛОЖЕНИЕ Б

Код контроллера для создания файлов

return Bookshelf

.transaction(transacting => {

return new FilesType[fileType](newFile)

.save(null, { transacting })

.then(createdFileType => new Files({

file\_id: createdFileType.id,

file\_type: req.file\_type,

user\_id: req.user.id,

workspace\_id: req.workspace.id

})

.save(null, { transacting })

.then(() => createdFileType)

)

})

.tap(createdFileType => Promise

.fromCallback(cb => fs

.readFile(`./public/${req.file\_type}/${createdFileType.get('name')}`, (err, data) => {

if (err)

throw err;

const base64data = new Buffer(data, 'binary');

return S3Service.put({

Key: createdFileType.get('name'),

Body: base64data,

Bucket: config.get('AWS.S3.bucket')

})

.then((res) => createdFileType.save({

s3\_key: res.ETag

}))

.then(() => cb())

.catch(cb);

})

)

)

.tap(createdFileType => res.status(201).send(createdFileType))

.catch(next);