КОНКУРСНАЯ РАБОТА

Студента института «Инфокомуникационных систем и технологий»   
направления «Прикладная информатика в системах управления»

Государственного бюджетного образовательного учреждения высшего образования Московской области «Технологический университет»

ИВАНОВА ИЛЬИ АЛЕКСАНДРОВИЧА

На тему:

ПОСТРОЕНИЕ РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ ДЛЯ ОПРЕДЕЛЕНИЯ АКТУАЛЬНОСТИ НАУЧНЫХ ПУБЛИКАЦИЙ

2020г

# СОДЕРЖАНИЕ

Оглавление

[СОДЕРЖАНИЕ 2](#_Toc52207508)

[ВВЕДЕНИЕ 3](#_Toc52207509)

[ОСНОВНАЯ ЧАСЬ 5](#_Toc52207510)

[1. ОБЩЕЕ ОПИСАНИЕ 5](#_Toc52207511)

[2. ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ 8](#_Toc52207512)

[2.1 ОСНОВНОЙ АЛГОРИТМ ФОРМИРОВАНИЯ РЕКОМЕНДАЦИЙ 8](#_Toc52207513)

[2.2 АЛГОРИТМ МЕТОДА ОПРЕДЕЛЕНИЯ КЛАССОВ. 10](#_Toc52207514)

[2.3 АЛГОРИТМ МЕТОДА ОПРЕДЕЛНИЯ АКТУАЛЬНОСТИ. 12](#_Toc52207515)

[2.4 АЛГОРИТМ МЕТОДА ФОРМИРОВАНИЯ РЕКОМЕНДАЦИИ. 13](#_Toc52207516)

[3. ОПИСАНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ. 14](#_Toc52207517)

[4. ПРИМЕР РАЗРАБОТКИ КЛИЕНСТКОЙ ЧАСТИ. 16](#_Toc52207518)

[ВЫВОДЫ. 19](#_Toc52207519)

[ЗАКЛЮЧЕНИЕ. 20](#_Toc52207520)

[СПИСОК ЛИТЕРАТУРЫ. 21](#_Toc52207521)

# ВВЕДЕНИЕ

Тематика рекомендательных систем и машинного обучения, применяемого в них, набирает популярность. С каждым днём мы можем наблюдать, как появляются всё новые и новые алгоритмы и системы, построенные на базе нейронных сетей с использованием машинного обучения. А также всё больше и больше инструментов, позволяющих создавать подобные решения.

Рекомендательные системы — это большой класс моделей, цель которых формирование рекомендаций пользователю в правильном месте и в правильное время. Это комплексы алгоритмов, программ и сервисов, основная задача которых предсказать, что будет интересно пользователю, имея информацию о его профиле, либо иные данные.

В качестве одного из примеров рекомендательной системы в современном представлении можно привести movielens.org[6,7], предлагающий пользователям фильмы на основе их предпочтений.

Этот сервис предоставляет всем желающим обширный набор данных о фильмах и рейтингах, поставленных им пользователями. Этот набор данных был использован в множестве исследований в области рекомендательных систем последних двух десятилетий.

Так же, на данный момент существует большое множество рекомендательных систем в большинстве социальных сетей, видео хостингов и сайтов. Так как очень многие компании, заинтересованы в том, чтобы привлекать пользователей и выдавать им, только то, что будет им интересно.

Данная тематика, постепенно начинает затрагивать многие сферы деятельности в нашей жизни. Одна из таких сфер – сфера образовательных услуг.

В каждом высшем учебном заведении, существует большой поток публикаций, в связи с чем, может возникнуть необходимость правильной оценки публикуемых статей. Одним из критериев такой оценки является – актуальность выбранной темы, на данный момент времени. В связи с этим, в рамках данной работы, будет рассмотрено решение, которое представляет собой, рекомендательную систему, способную не только определить актуальность публикуемой статьи, но и дать рекомендации по увеличению этой актуальности.

Существует большое количество инструментов, позволяющих обрабатывать и анализировать текст применяя методы машинного обучения и статистического анализа текста. В рамках описываемой в данной работе – системы, наиболее подходящим является такой инструмент как: Yet Another Keyword Extractor [1]. Представляющий собой – легкий, автоматический метод извлечения ключевых слов, который основан на статистических характеристиках текста, извлеченных из отдельных документов, для выбора наиболее важных ключевых слов текста.

За счёт того, что данный инструмент использует уже предобученную нейронную сеть – этот метод, можно охарактеризовать как неконтролируемый и автоматический.

Так же, существует такой инструмент как: BERT [2, 3, 4, 5]. Являющийся одним из методов обработки естественного языка, основанный на использовании нейронных сетей. Каждое слово кодируется отдельным числом. Это делается для того, чтобы нейросеть могла работать с поданным на вход закодированным текстом. На практике чаще применяют для одного слова не одно число - а вектор, содержащий в себе более одного числа. Для того, чтобы задать нейросети "ассоциации" со словом, рядом с ним по числовому значению, ставят близкое по смыслу слово или набор слов так, чтобы разность между числовыми значениями этих слов была минимальна.

Оба этих инструмента используют обработку естественного языка (Natural language processing, NLP). Данная обработка относится к области искусственного интеллекта, которая занимается лингвистикой, чтобы дать компьютерам возможность понять, естественный человеческий язык.

Важно отметить, что любая рекомендательная система, нуждается в обучении нейронной сети. Существует множество алгоритмов обучения и наиболее подходящим - для данной информационной системы является алгоритм обратного распространения ошибки.

Это алгоритм обучения нейронных сетей прямого распространения. Относится к методам обучения с учителем, поэтому требует, чтобы в обучающих примерах были заданы целевые значения.

В основе идеи алгоритма лежит использование выходной ошибки нейронной сети.

Алгоритм является итеративным и использует принцип обучения «по шагам», когда веса нейронов в сети корректируются после подачи на ее вход одного обучающего примера.

# ОСНОВНАЯ ЧАСЬ

## ОБЩЕЕ ОПИСАНИЕ

Основная цель разрабатываемой информационной системы состоит в том, чтобы, получив от пользователя публикацию, на основе её текста, определить и дать рекомендации по изменению содержания статьи с целью повышения её актуальности.

Важно отметить, что система, может осуществлять взаимодействие с пользователем, через веб - интерфейс. Данный интерфейс - может быть размещён как на учебном портале университета, так и иметь собственную, отдельную площадку. Такая возможность, существует благодаря разработке RESTful API в описываемой системе. Данный подход гарантирует дальнейшую масштабируемость взаимодействия компонентов информационной системы за счёт их независимого внедрения.



Схема 1. RESTful API сервис.

При загрузке научной статьи или публикации в систему, последняя в свою очередь – после обращения к RESTful API, сможет подсказать пользователю:

* Какие ключевые слова можно было бы добавить или удалить;
* К какой тематике относится его статья;
* Что можно было бы добавить или исправить в тексте научной публикации.

При несоответствии предложенного рекомендательной системой результата – с мнением пользователя – последний в свою очередь сможет внести коррективы в предложенный результат. Так же, планируется сбор информации об отклонении или принятии пользователем предложенных рекомендаций. 

Рисунок 2. Диаграмма BPMN.

Таким образом, рассматриваемая рекомендательная система будет постепенно обучаться, выдавая всё более точные варианты ответа, используя такой алгоритм обучения как «Метод обратного распространения ошибки» описанный в разделе «Введение».

Так же, необходимо первичное обучение сети, на примерах статей прошлых годов или статей из интернета. При этом, размер выборки должен быть не менее нескольких тысяч.

В процессе работы системы применяется такая технология как «Yet Another Keyword Extractor» так же, описанная ранее в разделе «Введение». Она позволит нам получать из текста публикации ключевые слова.

Важно отметить, что у Yet Another Keyword Extractor есть понятие величина важности ключевого слова. От неё зависит, какие найденные в результате анализа ключевые слова, попадут в результирующий список. На основании этой величины в списке найденных слов формируются кандидаты в результирующий список. Сам результирующий список представляет собой список пар, состоящих из найденного ключевого слова и значения его величины важности.

В базовом варианте Yet Another Keyword Extractor для настройки доступно несколько параметров:

1. language: отвечает за язык текста;
2. maxNgramSize: отвечает за максимальное количество слов, в ключевом предложении или словосочетании;
3. numberOfKeywords: определяет кол-во ключевых слов в результирующем списке. Данное значение в рамках данного решения имеет по умолчанию значение 10.

## 2. ИСПОЛЬЗУЕМЫЕ АЛГОРИТМЫ

### 2.1 ОСНОВНОЙ АЛГОРИТМ ФОРМИРОВАНИЯ РЕКОМЕНДАЦИЙ

1. На вход поступает публикация с выделенными в ней ключевыми словами.

Важно учитывать, что статьи должны включать разделы:

* Ключевые слова;
* Основной текст.

1. После загрузки статьи через интерфейс, происходит подготовка её текста к отправке в API – Yet Another Keyword Extractor, развёрнутому в Docker Container-е. Данная процедура - поможет нам определить предполагаемые ключевые слова, на основе текста статьи. На сервере, данная обработка начинается с этапа фильтрации текста за счёт использования инструмента языка Java для программной обработки документов Microsoft, носящего название – Apache POI [8]. Также, если на вход поступает публикация в формате файла PDF, будет использоваться такой инструмент, как IText [11], позволяющий получить текст публикации из файла формата PDF.

После выгрузки текста из файла, он проходит несколько этапов обработки:

* 1. Замена всех переносов параграфов в тексте на «.» для лучшего распознавания текста Yet Another Keyword Extractor-ом;
  2. Удаление англоязычных блоков текста;
  3. Осуществление поиска по всему тексту публикации, с целью выделения нужных для работы, его составных частей таких как:
     1. Блок текста содержащий название статьи;
     2. Блок текста содержащий ключевые слова;
     3. Блок текста содержащий ссылки на источники;
     4. Блок текста содержащий основной текст публикации;
  4. Из найденных частей текста, составляется готовый к анализу текст научной публикации, который будет содержать только необходимые нам блоки текста, загруженной научной публикации, что в свою очередь позволит нам, используя Yet Another Keyword Extractor — более точно определить ключевые слова анализируемой публикации.

1. На основании ключевых слов определяется принадлежность публикации к классам.
2. Выборка сохранённых значений актуальности по классам
3. Формирование рекомендации на основании полученных значений актуальности.

Так — же, разработаны вспомогательные алгоритмы:

1. Определения классов, по ключевым словам;
2. Определения актуальности по классам;
3. Формирования рекомендации по актуальности.

В графическом представлении весь алгоритм можно представить следующим образом:

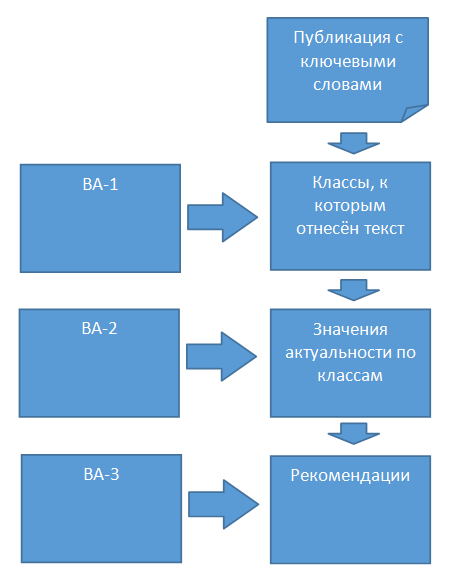


Схема 2. Основной алгоритм.

### 2.2 АЛГОРИТМ МЕТОДА ОПРЕДЕЛЕНИЯ КЛАССОВ.

Входные параметры: список (массив) ключевых слов.

Выходные параметры: список (массив) классов.

Логика работы:

Производится преобразование входного списка в выходной, при использовании вспомогательной таблицы №1. Столбцы таблицы – классы, строки таблицы – ключевые слова, значения не пересечении столбцов и строк – веса соответствия, по умолчанию равные нулю.

1. Формируется пустой список для классов.
2. Для каждого ключевого слова во входном списке ищется это ключевое слово в таблице.
   1. Если ключевые слова закончились, возвращается результирующий список классов.
   2. Если ключевого слова нет в таблице – добавляется новая строка, все значения в этой строке заполняются нулями. Осуществляется переход к следующему ключевому слову.
   3. Если ключевое слово есть, то находятся классы, для которых значения в соответствующей строке не равны нулю.
3. Для каждого найденного класса проверяется, есть ли он уже в списке классов.
   1. Если класса нет в списке, он сохраняется в базу данных, со значение из таблицы.
   2. Если класс есть в списке, увеличивается значение на значение из таблицы.

Для первоначального определения классов и их значений на пересечении с ключевыми словами используется алгоритм кластеризации *k*-средних.

Пример работы алгоритма:

Вспомогательная таблица №1:

Таблица 1.

|  |  |  |
| --- | --- | --- |
|  | Класс1 | Класс2 |
| Физика | 1 | 0 |
| Математика | 1 | 1 |

Входной список ключевых слов: {математика, химия}.

Выходной список классов после работы алгоритма: {Класс1 : 1, Класс2 : 1}.

Вспомогательная таблица №1 после работы алгоритма:

Таблица 2.

|  |  |  |
| --- | --- | --- |
|  | Класс1 | Класс2 |
| Физика | 1 | 0 |
| Математика | 1 | 1 |
| Химия | 0 | 0 |

### 2.3 АЛГОРИТМ МЕТОДА ОПРЕДЕЛНИЯ АКТУАЛЬНОСТИ.

Входные параметры: список (массив) классов.

Выходные параметры: список (массив) пар «класс-актуальность».

Логика работы:

Используется вспомогательная таблица №2, со столбцами «Класс» и «Актуальность».

1. Для каждого класса во входном списке в выходной добавляется - список пар «класс-актуальность».
2. Для формирования таблицы №2, используется алгоритм опроса поискового API Yandex.xml и Google Search API, которое по имени класса возвращает количество результатов поиска. Данное значение будет сохранено во вспомогательную таблицу №2 как значение актуальности класса. Данный алгоритм опроса реализован в виде независимой прослойки, что позволяет легко изменять принцип его работы и внедрять его в другие модули приложения.

Все вспомогательные таблицы, описанные в данных алгоритмах – расположены в СУБД MySQL. Доступ к ней осуществляется благодаря использованию Spring JDBC.

### 2.4 АЛГОРИТМ МЕТОДА ФОРМИРОВАНИЯ РЕКОМЕНДАЦИИ.

Входные параметры: список (массив) пар «класс-актуальность».

Выходные параметры: текст рекомендации.

Логика работы:

1. Необходимо начать с пустого текста рекомендации.
2. Осуществляется сравнение актуальности каждого класса из входного списка с пороговым значением
3. Если пороговое значение превышено, добавляется текст формата «класс – актуальность» к тексту рекомендации. По умолчанию пороговое значение равно 0.

Важно отметить, необходимость сделать редактируемым на этапе обучения — список получаемых из Yet Another Keyword Extractor ключевых слов, с возможностью корректировки как самого ключевого слова, так и значения его «важности».

Так же предусмотрена возможность добавления и удаления ключевых слов в случае, если Yet Another Keyword Extractor даст не корректный результирующий список.

После внесения всех необходимых изменений, предусмотрено сохранение результатов анализа классов и процесса выделения ключевых слов ­– в базу данных, вместе с привязкой к тексту публикации и параметрами Yet Another Keyword Extractor, использованными для получения ключевых слов. А также сохранение списка пар класс-актуальность и самой сформированной рекомендации.

Описанные алгоритмы реализованы в виде независимых программных модулей, что позволяет в дальнейшем, использовать их в других подобных разработках, при этом осуществлять их настройку, не затрагивая при этом основные компоненты приложения, использующие их.

## 3. ОПИСАНИЕ ТЕХНИЧЕСКИХ СРЕДСТВ.

Для реализации вспомогательных таблиц была выбрана СУБД MySQL.

Структура базы данных:

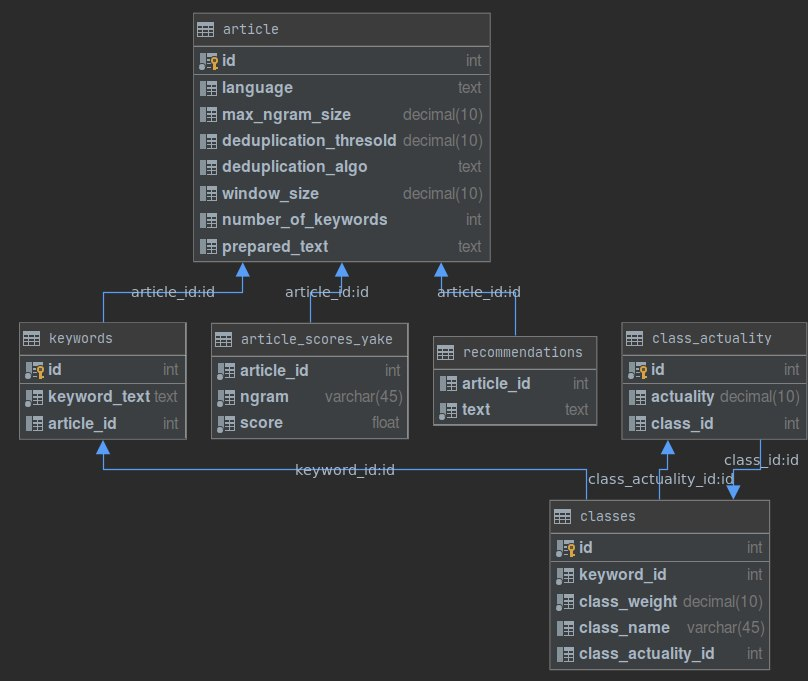


Рисунок 3. Схема структуры базы данных.

В качестве основного языка приложения, был выбран язык Java и в следствие принятия решения о разработке веб сервиса, были выбраны Spring Boot, Spring JDBC

Используемые зависимости:

1. Обработка текста:
   1. Apache POI [8];
   2. IText [11].
2. Запросы к Yet Another Keyword Extractor:
   1. OKHTTP3 [9].
3. Миграции и версионность базы данных:
   1. Flyway [10].

На клиентской части приложения, были использованы такие инструменты как Vue.js, Vuex, TypeScript, Axios и шаблонизатор Twig.

Также, планируется полноценная реализация с помощью Quasar Framework, для создания приложения по технологии SPA (Single Page Application).

Система запущена на серверной виртуальной машине, под управлением Ubuntu 18.04, 4GB RAM, Intel Xeon с выделенными 4-мя ядрами. В процессе опытной эксплуатации, доказано, что данные характеристики полностью удовлетворяют ресурсо-затратам описываемой информационной системы и подходят в качестве рекомендованных системных требований.

Минимальные системные требования:

* Intel core i3;
* 4GB RAM DDR3;
* 35МБ свободного места на носителе.

Структура компонентов системы:



Рисунок 4. Диаграмма развёртывания.

## 4. ПРИМЕР РАЗРАБОТКИ КЛИЕНСТКОЙ ЧАСТИ.

В качестве примера рассмотрим реализацию с помощью Quasar Framework.

В интерфейсе системы, было решено реализовать поэтапное формирование рекомендаций как на Рисунке 5. При этом, реализовав на промежуточных этапах служебные интерфейсы такие как:

1. Формирование входных параметров Yet Another Keyword Extractor;
2. Определение Ключевых слов;
3. Анализ актуальности.

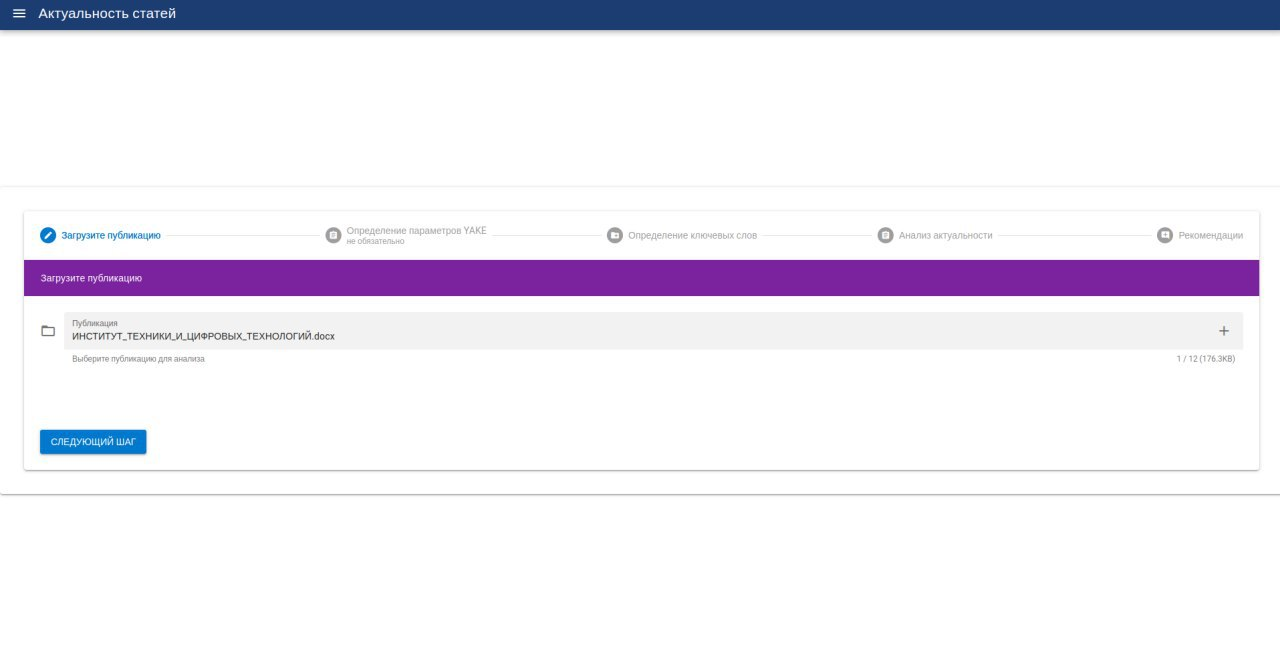


Рисунок 5. Этапы формирования рекомендаций.

Данные этапы включают в себя:

1. Загрузка публикации;
2. Формирование входных параметров Yet Another Keyword Extractor (Служебный);
3. Определение и редактирование ключевых слов (Служебный);
4. Анализ актуальности (Служебный);
5. Получение рекомендаций.

Функционал этапов:

1. Загрузка файла публикации;
2. Формирование входных параметров Yet Another Keyword Extractor;
3. Определение ключевых слов;
   1. Редактирование предложенного списка ключевых слов;
   2. Добавление новых ключевых слов;
   3. Удаление ключевых слов;
   4. Сохранение списка ключевых слов и параметров Yet Another Keyword Extractor c файлом статьи.
4. Анализ актуальности;
   1. Выдача списка пар Класс – Актуальность.
5. Формирование рекомендаций;
   1. Формирование общих рекомендаций;
   2. Формирование рекомендаций по редактированию ключевых слов в тексте публикации;
   3. Внесение изменений в рекомендации;
   4. Вывод значения актуальности;
   5. Принятие или отклонение рекомендаций.

На этапе формирования рекомендации, существует упомянутая выше – возможность принять или отклонить рекомендацию, а также внести коррективы в предложенный результат. Пример – Рисунок 6. Это также необходимо для сбора информации и переобучения нейронной сети.

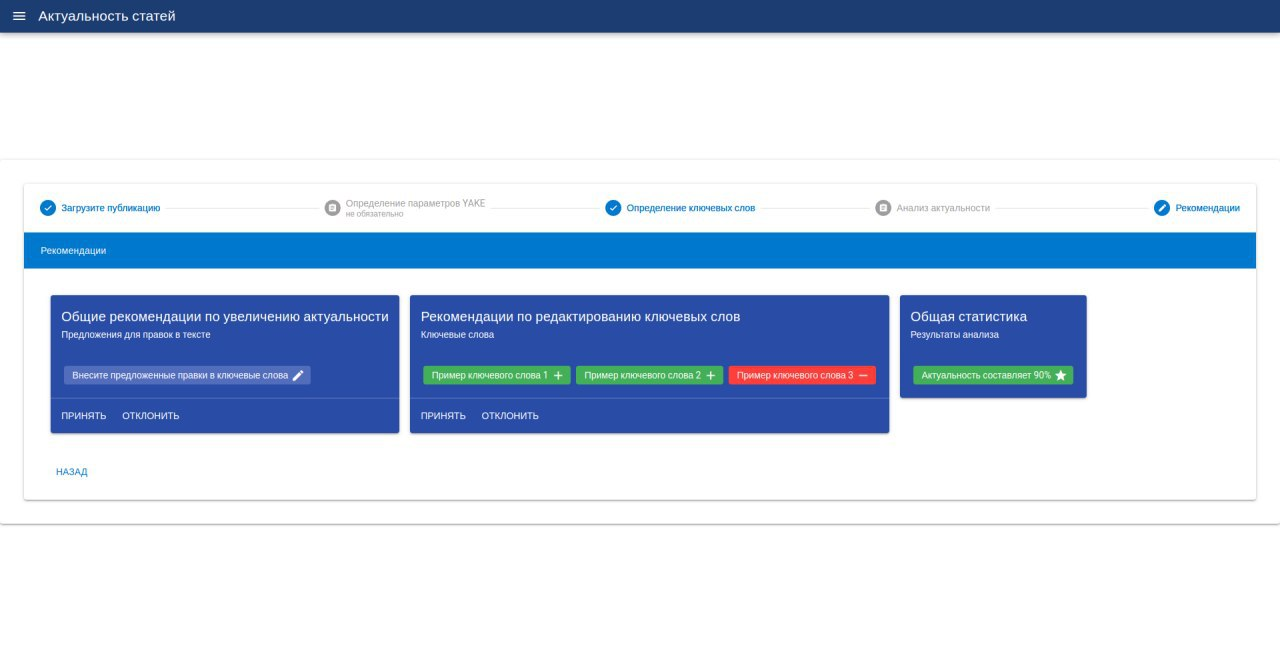


Рисунок 6. Пример сформированной рекомендации.

# ВЫВОДЫ.

В процессе работы над рассматриваемой системой, была изучена предметная область, выявлены уже существующие рекомендательные системы и определена актуальность тематики рекомендательных систем – во многих областях, в том числе в сфере образовательных услуг. Исходя из этих данных и результатов проделанной на данный момент времени работы – можно сказать, что у описанной системы – не существует аналогов, так-как тематика рекомендательных систем в сфере образовательных услуг – находится на ранних стадиях развития.

Установлено, что – существует множество современных инструментов для обработки и анализа текста, таких как: Apache POI, IText, Yet Another Keyword Extractor, способные помочь и существенно упростить разработку подобных систем и сервисов.

# ЗАКЛЮЧЕНИЕ.

Информационная система, описанная в рамках данной работы, относится к классу рекомендательных систем. Исходя из этого, мы получаем готовый инструмент - для статистического анализа текста публикаций и формирования рекомендаций, помогающих пользователям на их основе, принимать необходимые решения по редактированию содержания их публикаций, что как следствие – поможет увеличить их актуальность.

За счёт функционала и выбранных инструментов, данная система легко расширяема и обладает достаточной гибкостью при настройке отдельных её модулей, которые могут в дальнейшем быть использованы как микросервисы.

Реализация RESTful API - позволит использовать описываемую систему - с любыми интерфейсами и в любых проектах, начиная с приложений под различные операционные системы и заканчивая веб – приложениями и приложениями под мобильные устройства.

В настоящее время - описываемая информационная система, находится на стадии опытной эксплуатации в ГБОУ ВО МО «Технологический университет».

# СПИСОК ЛИТЕРАТУРЫ.

1. Yet Another Keyword Extractor (Yake). Unsupervised Approach for Automatic Keyword Extraction using Text Features. [Электронный ресурс] // URL: <https://github.com/LIAAD/yake> (дата обращения: 27.08.2020).
2. A Visual Guide to Using BERT for the First Time. [Электронный ресурс] // URL: <https://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/> (дата обращения: 14.07.2020).
3. BERT Word Embeddings Tutorial. [Электронный ресурс] // URL: <https://mccormickml.com/2019/05/14/BERT-word-embeddings-tutorial/> (дата обращения: 17.07.2020).
4. Keyphrase Extraction using SciBERT. [Электронный ресурс] // URL: <https://github.com/pranav-ust/BERT-keyphrase-extraction> (дата обращения: 19.08.2020).
5. BERT. [Электронный ресурс] // URL: https://github.com/google-research/bert (дата обращения: 12.07.2020).
6. MovieLens Non-commercial, personalized movie recommendations. [Электронный ресурс] // URL: https://movielens.org/ (дата обращения: 03.08.2020)
7. MovieLens Unplugged: Experiences with an Occasionally Connected Recommender System. [Электронный ресурс] // URL: http://files.grouplens.org/papers/miller-iui03.pdf (дата обращения: 02.09.2020).
8. Apache POI. [Электронный ресурс] // URL: https://poi.apache.org (дата обращения: 09.07.2020).
9. OkHTTP. [Электронный ресурс] // URL: https://square.github.io/okhttp (дата обращения: 11.07.2020).
10. FlyWay. [Электронный ресурс] // URL: https://flywaydb.org/documentation (дата обращения: 19.07.2020).
11. IText. [Электронный ресурс] // URL: https://itextpdf.com/ru (дата обращения: 21.07.2020).