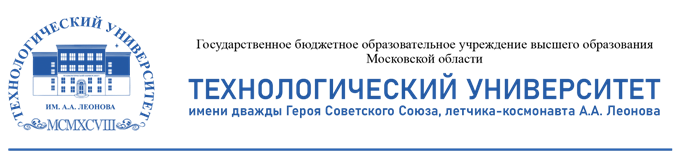
****

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ**

**ОТЧЕТ**

о прохождении преддипломной практики

Студента группы ИО-17 4 курса

направление подготовки (специальность) Прикладная информатика

Фамилия Иванов

Имя Илья

Отчество Александрович

Место прохождения практики ГБОУ ВО МО «Технологический университет» (Учебно-научной лаборатории интеллектуальных систем управления)

Время прохождения практики

Руководитель практики от кафедры (факультета)

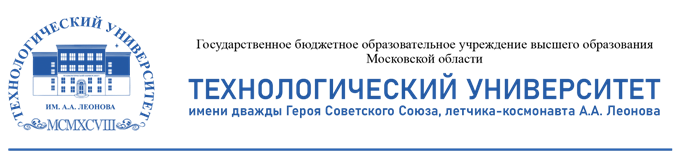
(Фамилия, им, отчество, должность) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики от организации, (предприятия, учреждения)

(Фамилия, им, отчество, должность) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Заведующий кафедрой

Королев 2021 г.

**

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ**

**ЗАДАНИЕ**

**НА ПРЕДДИПЛОМНУЮ ПРАКТИКУ**

Выдано студенту Иванову Илье Александровичу 4 курса, группы ИО-17,

(Ф.И.О., курс, группа)

РАБОЧАЯ ПРОГРАММА ПРАКТИКИ

ГБОУ ВО МО «Технологический университет» (Учебно-научной лаборатории интеллектуальных систем управления)

(наименование организации)

1. Цель и задачи практики

2. Ведение и оформление дневника практики.

3. Составление и оформление отчета по практике.

4. Индивидуальное задание по теме практики:

Разработка и тестирование рекомендательной системы для определения актуальности научных публикаций.

Начало практики «5» апреля 2021 г

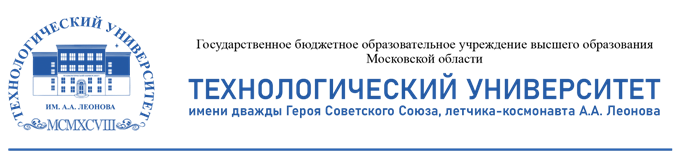
Конец практики «16» мая 2021 г

Задание выдал\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О. руководителя от института)

Задание принял\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О. студента)



**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И УПРАВЛЯЮЩИХ СИСТЕМ**

**ДНЕВНИК**

**ПРОХОЖДЕНИЯ ПРЕДДИПЛОМНОЙ ПРАКТИКИ**

Фамилия, имя, отчество студента Иванов Илья Александрович

направление подготовки (специальность) Прикладная информатика

Специализация Прикладная информатика в системах управления

Курс 4 Группа ИО-17

Руководитель практики от кафедры (факультета)

Организация (предприятие, учреждение)

Руководитель практики от организации (предприятия, учреждения)

**Особые отметки**

Выбыл на практику « » 20 г.

Руководитель практики от кафедры (факультета)

Прибыл в организацию (предприятие) « » 20 г.

Руководитель практики от организации (предприятия)\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выбыл из организации (предприятия) « »\_\_\_\_\_\_\_ 20 г.

Руководитель практики от организации (предприятия)

Прибыл в институт « » 20 г.

Руководитель практики от кафедры (факультета)

|  |  |  |  |
| --- | --- | --- | --- |
| **Месяц и число** | **Подразделение предприятия** | **Краткое описание выполненной работы** | **Подпись руководителя практики** |
| **1** | **2** | **3** | **4** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Начало практики 5 апреля 2021 Конец практики 16 мая 2021

Подпись практиканта\_\_\_\_\_\_\_\_\_\_\_\_\_

Содержание и объем выполненных работ подтверждаю.

Руководитель практики от предприятия \_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (Ф.И.О.)

М.П.

**Отзыв**

**на отчет по преддипломной практике**

Студента 4 курса группы ИО-17

Иванов Илья Александрович

(Ф.И.О.)

Место прохождения практики: ГБОУ ВО МО «Технологический университет» (Учебно-научной лаборатории интеллектуальных систем управления)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики

от предприятия \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (Ф.И.О.)

М.П.

Оглавление

[1. Введение. 7](#_Toc71258037)

[2. Описание общего алгоритма работы системы. 8](#_Toc71258038)

[3. Существующие рекомендательные системы. 10](#_Toc71258039)

[4. Описать базовые термины и определения в области машинного обучения и рекомендательных систем. 12](#_Toc71258040)

[4.1. Рекомендательные системы. 12](#_Toc71258041)

[4.2. Понятие нейронной сети и описание алгоритма её работы. 13](#_Toc71258042)

[4.3. Метрики. 17](#_Toc71258043)

[4.3.1. Метрики классификации. 17](#_Toc71258044)

[4.3.2. Метрики регрессии (MSE, MAE). 18](#_Toc71258046)

[4.4. Выборки. 19](#_Toc71258047)

[4.5. Архитектуры нейронных сетей. 20](#_Toc71258048)

[5. Преимущества разрабатываемой системы 22](#_Toc71258049)

[6. Архитектура сети и метрики определения качества обучения. 23](#_Toc71258050)

[7. Результаты разработки. 24](#_Toc71258051)

[10.1 Алгоритмы (Back-end). 24](#_Toc71258052)

[10.1.1 Алгоритм фильтрации. 24](#_Toc71258053)

[10.1.2 Алгоритм опроса поискового API. 25](#_Toc71258054)

[10.1.3 Алгоритм определения классов. 27](#_Toc71258055)

[10.1.4 Алгоритм нормирования значений. 28](#_Toc71258056)

[10.1.5 Алгоритм формирования рекомендаций. 29](#_Toc71258057)

[10.2. Пользовательский интерфейс (Front-end). 30](#_Toc71258058)

[8. Заключение 34](#_Toc71258059)

[9. Список используемых источников 35](#_Toc71258060)

# Введение.

Цель данной работы:

Разработка и тестирование рекомендательной системы для определения актуальности научных публикаций.

Задачи работы:

1. Описание общего алгоритма работы системы;
2. Рассмотреть существующие рекомендательные системы;
3. Описать базовые термины и определения в области машинного обучения и рекомендательных систем;
4. Определить преимущества разрабатываемой системы;
5. Определить архитектуру сети и метрики определения качества обучения;
6. Описать результаты разработки;
   1. Алгоритмы (Back-end);
   2. Пользовательский интерфейс (Front-end).
7. Составить общее заключение.

Задачи выполняются на основе приобретённых знаний умений и навыков. С применением инструментальных средств разработки, отладки и тестирования. А также, устройств для развёртывания.

# Описание общего алгоритма работы системы.

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 [1], развёрнутому в Docker Container-е. Данная процедура - поможет 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 [4]. Также, если на вход поступает публикация в формате файла PDF, будет использоваться такой инструмент, как IText [7], 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. Полученные ключевые слова проходят фильтр, который позволит извлечь только те слова, которые наиболее точно соответствуют авторским.
2. На основании ключевых слов определяется принадлежность публикации к классам.
3. Выборка сохранённых значений актуальности по классам.
4. Формирование рекомендации на основании полученных значений актуальности.

# Существующие рекомендательные системы.

В качестве одного из примеров рекомендательной системы в современном представлении можно привести movielens.org [2,3], 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.

Уже существуют системы – способные помогать студентам получать информацию из записанных преподавателями лекций в видео формате – в текстовом виде. Это достигается за счёт использования технологий и инструментов распознавания речи – способных анализировать аудиоматериалы, распознавать в них человеческую речь и переводить её в текстовый формат представления данных.

В качестве одного из примеров таких систем может выступать CoursEra [8] Это платформа онлайн обучения, основанная на рекомендательных системах. Она советует пользователям нужные и интересные для них курсы, способные привлечь их внимание.

Также в качестве примера рекомендательной системы в сфере образовательных услуг можно описать MathGarden [9] – работающим с конкретными задачами для обучающихся. Он представляет собой тренажёр по арифметике для учеников начальных школ. Рекомендательная система этого ресурса – способна предложить ученику задачи – оптимально подобранные по текущему уровню освоения учеником учебной программы.

# Описать базовые термины и определения в области машинного обучения и рекомендательных систем.

## **4.1. Рекомендательные системы.**

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.

Обработка естественного языка

Общее направление искусственного интеллекта и математической лингвистики направленные на работу с информацией в текстовом представлении.

## **4.2. Понятие нейронной сети и описание алгоритма её работы.**

Сеть представляет собой структуру связанных между собой нейронов (Рис. 4.2.1.). Каждая связь между нейронами представляет собой некое число, называемое весом данной связи. Обычно данная величина находится в интервале [0,1].

На вход сети подаётся нужный набор данных в цифровом представлении. Данный набор будет продвигаться через связи – по сети, при этом умножаясь на соответствующий вес связи.



Рис. 4.2.1. Модель искусственного нейрона.

У каждого нейрона есть функция, которая определяет выходное значение нейрона исходя из взвешенной суммы выходов и порогового значения. Исходя из поставленной задачи необходимо выбрать тип функции активации.

Следует выбирать ту функцию, которая аппроксимирует искомую функцию лучше и способствует более быстрому обучению.

Всего различают типы:

1. Ступенчатая функция активации (Рис. 4.2.2.):

Функция А = активирована, если X> порогового значения, иначе нет.

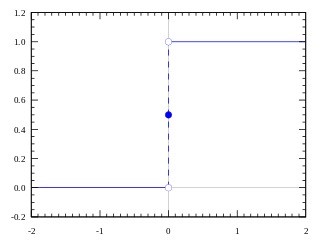


Рис. 4.2.2. Ступенчатая функция активации.

1. Линейная функция активации (Рис. 4.2.3.):

В отличии от ступенчатой функции – позволяет получать промежуточные значения активационной функции.

*Имеет формулу*: A = cx

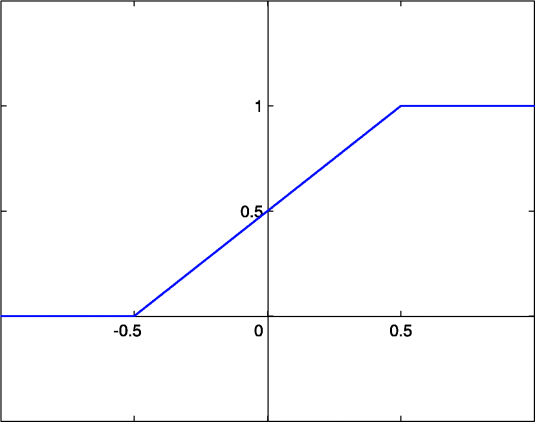
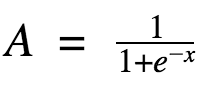


Рис. 4.2.3. Линейная функция активации.

1. Сигмоида (Рис. 4.2.4.):

*Имеет формулу*:

*Имеет следующие преимущества*:

* + Нелинейна;
  + Не бинарна.

*Недостатки*:

* При приближении значения Х к одной из горизонтальных частей кривой – обучение может очень сильно замедлиться.

Числа Y в диапазоне значений X от -2 до 2 меняются очень быстро. Это может означать, что при малейшем изменении X произойдёт существенное изменение Y.

Данная функция чаще всего применяется в задачах классификации стремясь привести значение либо к 1, либо к 0.

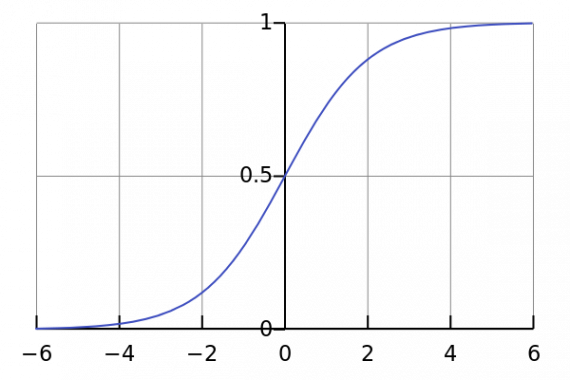
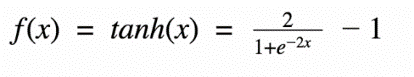
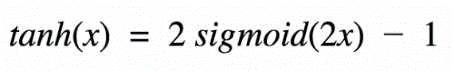


Рис. 4.2.4. Сигмоида.

1. Гиперболический тангенс (Рис. 4.2.5.):

*Имеет формулу:*

Является скорректированной версией сигмоидной функции активации.



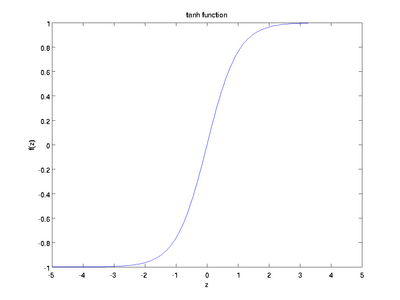


Рис. 4.2.5. Гиперболический тангенс.

1. ReLu (Рис. 4.2.6.):

*Имеет формулу: A(x) = max(0,x)*

*Возвращает Х если Х>0 и 0 если X<0*

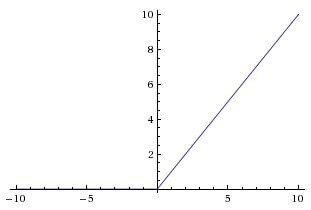


Рис. 4.2.6. ReLu.

## **4.3. Метрики.**

### **4.3.1. Метрики классификации.**

#### **4.3.1.1. Точность классификации.**

Базовая метрика, определяющаяся формулой:

Где n – количество правильных прогнозов, N – общее количество прогнозов.

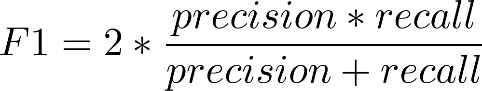
#### **4.3.1.2. Точность**

### 

**Определяющаяся формулой:**

**Где** TP — истинно-положительное решение, FP — ложноположительное решение;

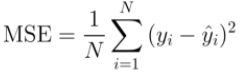
#### **4.3.1.3. F-1 мера.**

****

**Определяющаяся формулой:**

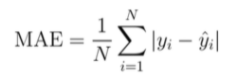
### **4.3.2. Метрики регрессии (MSE, MAE).**

#### **4.3.2.1. Mean Squared Error (MSE) Среднеквадратичная ошибка.**

Одна из самых популярных метрик в задачах регрессии. Находит ошибку между прогнозируемым и полученным значением.

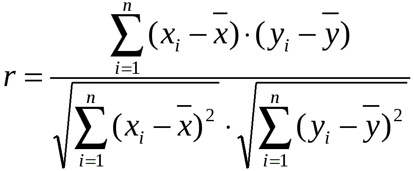
Определяется формулой:

#### **4.3.2.2 Mean Absolute Error (MAE) Средне абсолютная ошибка.**

Определяет среднее абсолютное расстояние между прогнозируемым и целевым значением.

Определяется формулой:

#### **4.3.2.3. Статистические метрики (корреляция).**

Используется для выявления зависимостей в наборе данных.

Определяется формулой:

## **4.4. Выборки.**

**Обучающая.**

Представляет собой набор данных, на которых будет происходить обучение нейронной сети. Используется для корректировки весовых значений.

**Валидационная.**

Набор данных, используемый для объективной оценки соответствия модели обучающей выборке при настройке гиперпараметров модели.

**Тестовая.**

Набор данных, используемый для объективной оценки окончательной модели.

## **4.5. Архитектуры нейронных сетей.**

**Многослойный перцептрон.**

Каждый узел в слое соединен с каждый узлом в последующем слое, это делает сеть полно связанной.

Применяется в задачах распознавания речи и машинном переводе.

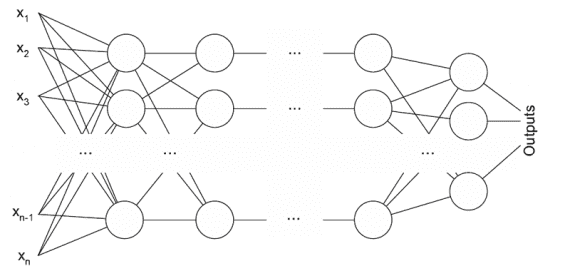


Рис 7.1 Схема архитектуры многослойный перцептрон.

**Свёрточная нейронная сеть.**

Состоит из одного или более сверхточных слоёв. Сами слои используют операцию свёртки для входных данных, а полученные результаты передают в следующий слой, что позволяет сети иметь более глубокую архитектуру при этом имея меньшее количество гиперпараметров.

Применяются в задачах распознавания изображений и видео контента.

**Рекурсивная нейронная сеть.**

Рекурсивно использует один и тот же набор весов, для достижения более структурированного предсказания.

Исходя из приведённых выше архитектур – установлено, что для них существуют следующие типы алгоритмов обучения:

**Обучение с учителем**.

Предполагает наличие заранее размеченного набора данных с уже известными ответами.

Наличие полностью размеченного набора данных означает, что каждому примеру в обучающем наборе соответствует ответ, который алгоритм должен получить на выходе, в качестве результата.

Обучение с учителем применяется для решения следующих задач:

1. Классификации;
2. Регрессии.

В задачах классификации алгоритм предсказывает дискретные значения, которые соответствуют номерам классов, принадлежащих к объектам. Качество алгоритма оценивается величиной точности и правильности классификации.

**Обучение без учителя.**

Задача данного подхода заключается в том, чтобы в наборе данных найти заранее не известные ответы без каких-либо указании путём поиска корреляции в данных.

Применяется в кластеризации и в нейронных сетях архитектуры «Автокодировщик».

# Преимущества разрабатываемой системы

Система позволяет не только определить актуальность публикуемой статьи, но и дать рекомендации по увеличению этой актуальности.

В отличии от множества других рекомендательных систем описанных выше – данная рекомендательная система не направлена напрямую на улучшение образовательного процесса, но призвана помочь преподавателям и студентам получать информацию по их статьям, способную поднять университет в мировых рейтингах на несколько строчек выше – за счёт рейтинга цитируемости университетов QS.

QS – это общемировой рейтинг университетов мира основанный на количестве цитирования статей. При составлении данного рейтинга процент учёта цитируемости статей составляет 40%, что является весомым фактором, позволяющим утверждать, что описываемая система – позволит увеличивать индекс цитируемости за счёт рекомендаций по увеличению актуальности научных статей и продвигать университет в мировых строчках рейтингов. Также это позволит ВУЗ-у привлекать новых специалистов и учёных, что улучшит образовательный процесс и его качество.

Преподаватели и студенты, использующие данную систему, смогут получать дополнительные возможности повышения своей квалификации, а также получать гранты и награды, за свои научные работы.

Хотелось бы отметить также, немаловажный фактор – сохранения времени, которое без использования данной системы – научные сотрудники тратят на поиск информации о источниках статей, анализу цитируемости и актуальности, а также на время, уходящее на редактирование статей в целях поднятия её описываемых характеристик.

# Архитектура сети и метрики определения качества обучения.

После изучения и анализа описываемых в П.7. архитектур и метрик – для решения задачи классификации в анализе текстов была выбрана архитектура нейронной сети «Автокодировщик».

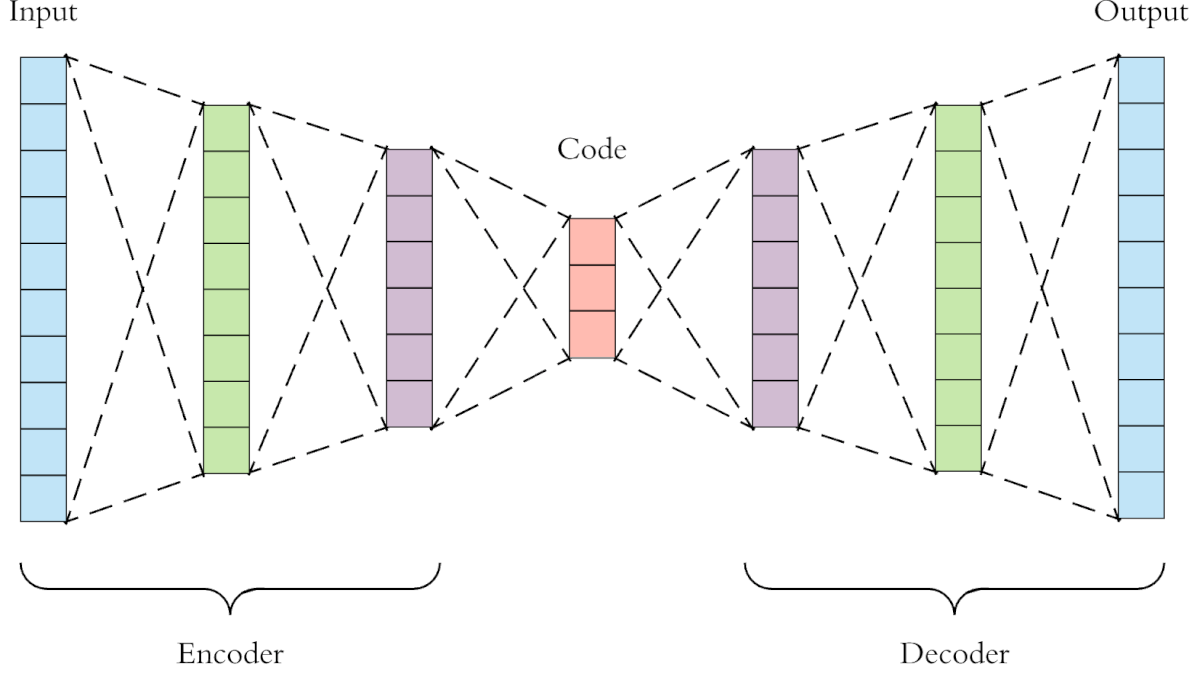


Рисунок 9.1 Структура сети «Автокодировщик».

Данная архитектура нейросети – позволяет применять обучение без учителя с применением алгоритма обратного распространения ошибки.

Также, данная архитектура выбрана благодаря тому, что позволяет эффективно работать с текстовым представлением информации. В результате множества проведённых экспериментов, данная архитектура очень хорошо проявила и зарекомендовала себя.

Автокодировщики используют данные на входе, кодируют и предпринимают попытки воссоздания исходных данных из результирующего кода. При изменении количество и размеров слоёв, автокодировщики могут повышать качество медиа контента, такого как: изображения и видео.

В качестве метрик необходимо использовать среднеквадратичную ошибку MSE. Данный вывод сделан, на основе рассмотренных в П.5.2. метрик.

# Результаты разработки.

## **10.1 Алгоритмы (Back-end).**

### **10.1.1 Алгоритм фильтрации.**

Используется для фильтрации ключевых словосочетаний полученных из Yet Another Keyword Extractor и позволяет понять - какие из ключевых словосочетаний пригодны для дальнейшей обработки.

1. На вход подаётся набор ключевых словосочетаний.
2. Для каждого отдельного слова вычисляются цифровые представления сохраняя связи между словами словосочетания.
3. Цифровые представления подаются на вход нейронной сети, которая на выходе сгенерирует результирующие цифровые представления слов.
4. Необходимо взять среднеквадратичную ошибку по каждому из них, для определения степени уверенности сети в полученном результате. Таким образом, формируется степень уверенности для каждого словосочетания из общего набора.
5. Далее определяем пороговое значение с помощью среднеарифметического по полученным величинам степеней уверенности для всего входного множества ключевых словосочетаний.
6. Перед формированием результирующего набора вида (ключевое словосочетание, прошло/нет фильтр) пропускаем каждое словосочетание через рассчитанное пороговое значение.

### **10.1.2 Алгоритм опроса поискового API.**

Пусть каждый класс имеет своё значение актуальности. Тогда путём опроса поискового API можно определить количество результатов поиска, и сохранить как значение актуальности для каждого из классов.

Данный алгоритм каждые 24 часа запрашивает количество результатов поиска для каждого из классов.

Затем происходит добавление новых значений актуальности в базу данных.

Это необходимо для того, чтобы можно было отслеживать тренды и динамику изменения актуальностей у классов (Рис.10.1.2.6.).



Блок схема 10.1.2.1 Алгоритм опроса поискового API.

### **10.1.3 Алгоритм определения классов.**

Для каждого из из множества ключевых слов N вычисляется его числовое представление.

Путём кластеризации множества ключевых слов, определяются классы, к которым они принадлежат.



Блок схема 10.1.3.1 Алгоритм определения классов.

Путём запроса в модуль NLP, вычисляются всевозможные расстояния от ключевого слова до каждого из классов.

### **10.1.4 Алгоритм нормирования значений.**

Для нормировки больших чисел и приведения их к интервалу [0..1] используется следующий алгоритм:

Определяется набор классов , где , для каждого из которых путём запроса в поисковй API определена актуальность .

Тогда, для определения актуальности набора ключевых слов , где , используется следующий алгоритм:

1. Нормируются актуальности , при помощи функции softmax:
2. Для каждого ключевого слова вычисляютяс расстояния от него до каждого из классов:
3. Нормируются расстояния для каждого ключевого слова :

1. Актуальность ключевого слова вычисляется следующим образом:

Итого, значения нормированных актуальностей классов принадлежат отрезку [0,1], как и значения нормированных расстояний до классов для каждого ключевого слова.

Так как, суммы нормированных значений равны 1 их можно трактовать как вероятности.

Актуальности ключевых слов также принадлежат отрезку [0,1], они больше, если больше актуальности классов, расстояние до которых меньше.

### **10.1.5 Алгоритм формирования рекомендаций.**

Перед формированием рекомендаций, необходимо сравнить актуальность каждого класса с пороговым значением.

Если актуальность выше – добавляем текст «класс – актуальность» к тексту рекомендации.

## **10.2. Пользовательский интерфейс (Front-end).**

В качестве примера рассмотрим реализацию с помощью Quasar Framework.

В интерфейсе системы, увидеть форму авторизации, а также страницу с функционалом для загрузки публикации и отображений рекомендаций.

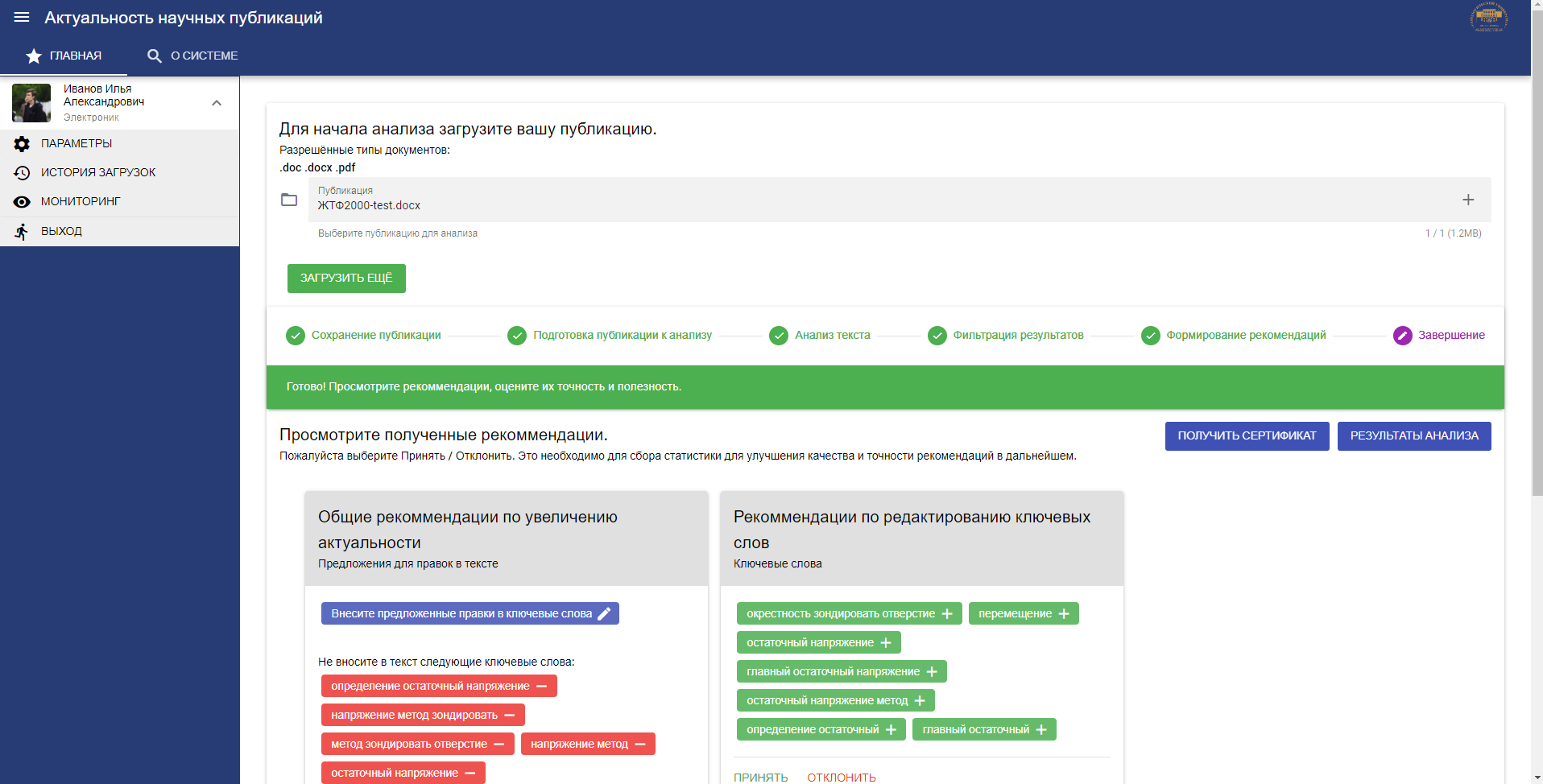


Рис.10.1.2.1 Главная страница с загруженной публикацией и сформированной рекомендацией.

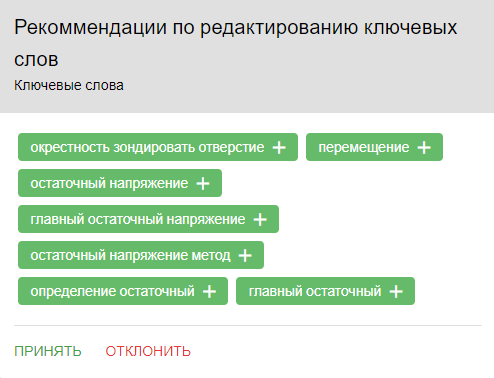


Рис.10.1.2.2. Главная страница с загруженной публикацией и сформированной рекомендацией.

На этапе формирования рекомендации, существует упомянутая выше – возможность принять или отклонить рекомендацию, а также внести коррективы в предложенный результат. Пример – Рисунок 2. Это также необходимо для сбора информации и переобучения нейронной сети.

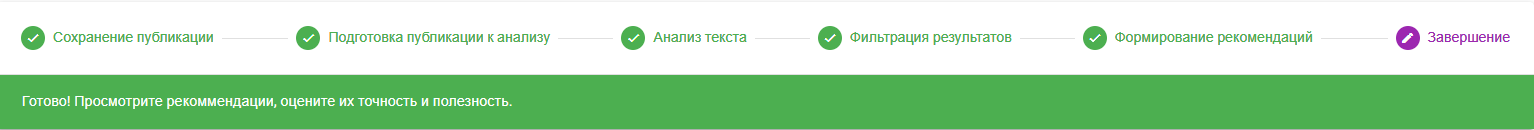


Рис.10.1.2.3. Этапы формирования рекомендации в реальном времени.

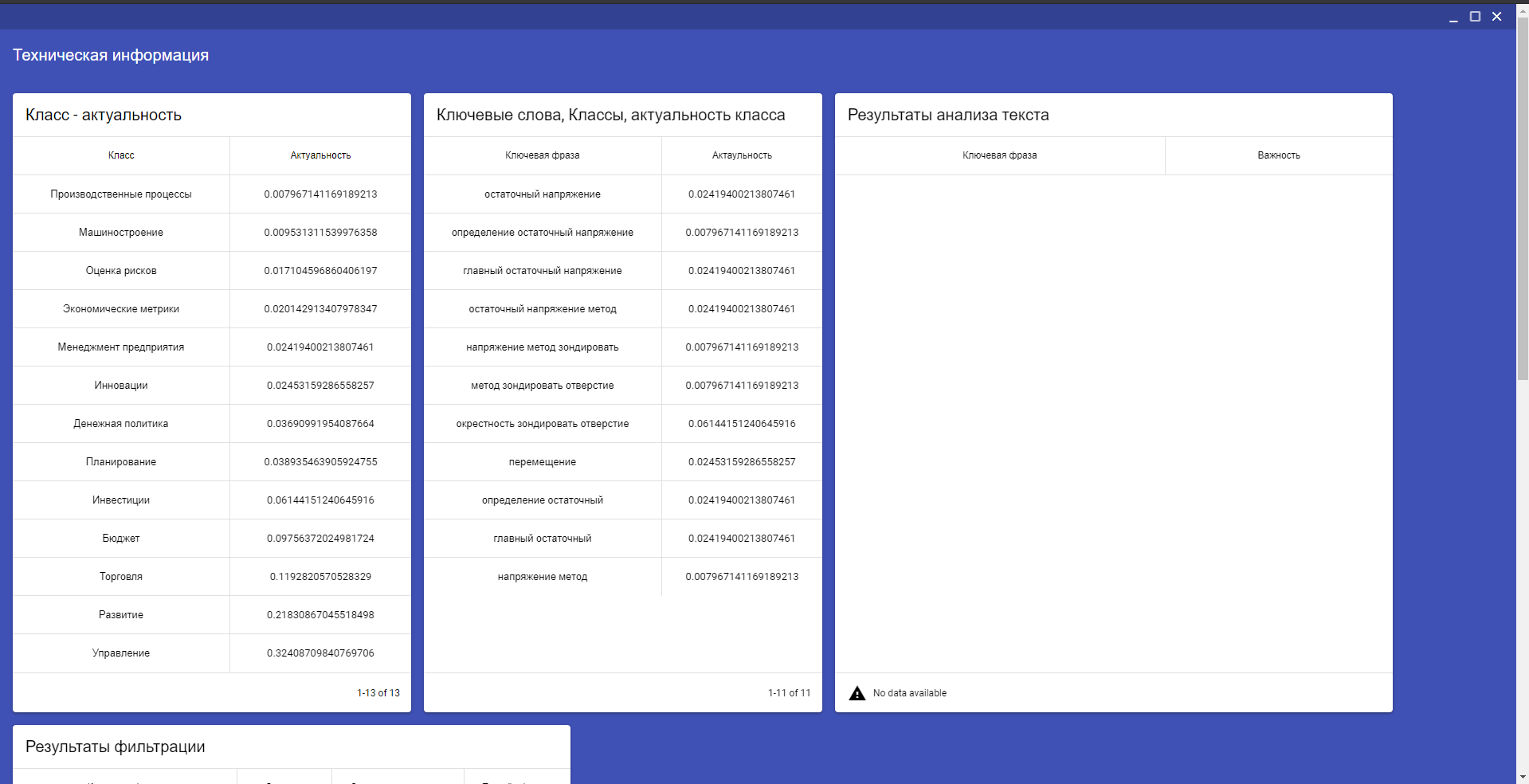


Рис.10.1.2.4. Пример статистических данных по анализируемой статье.

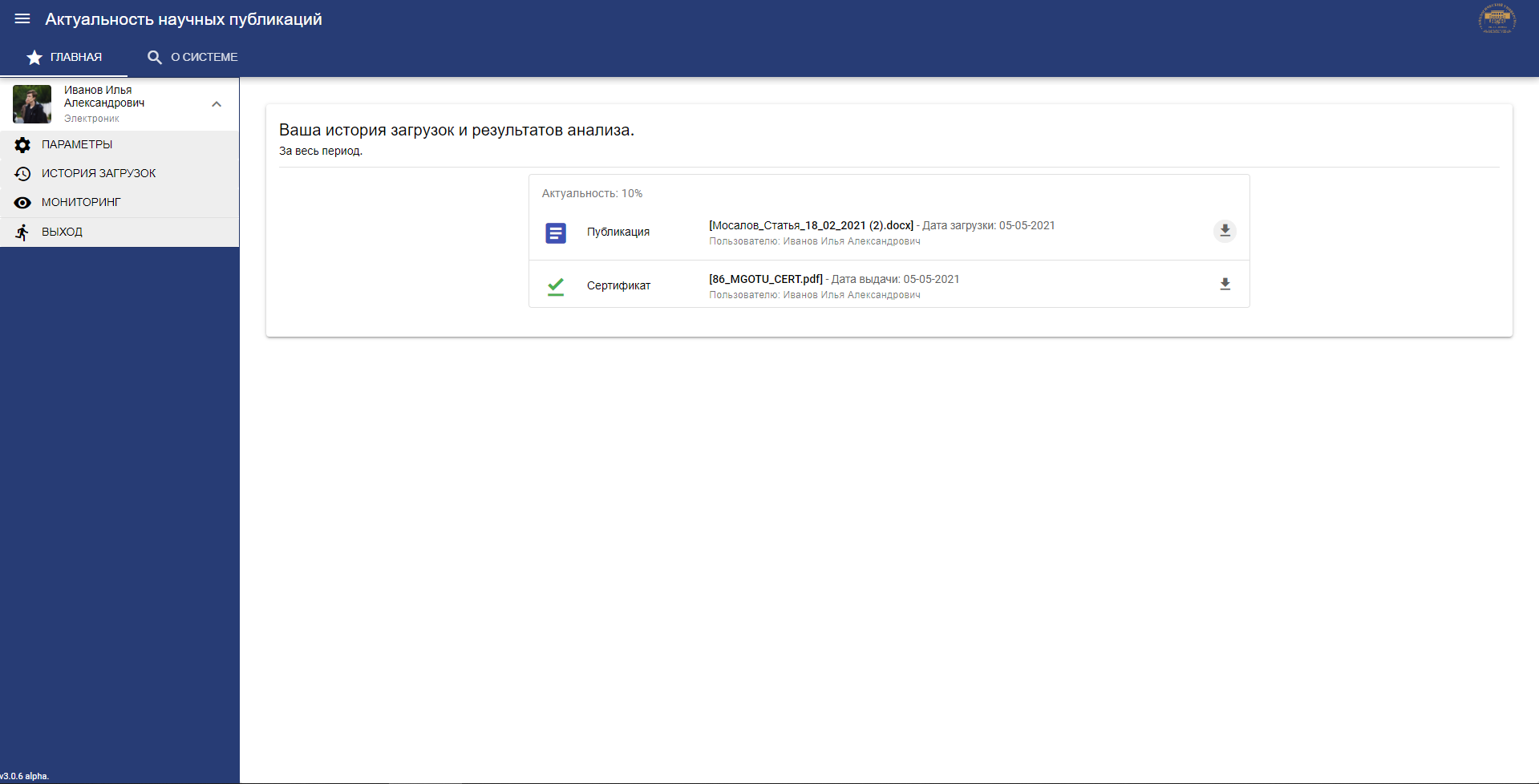


Рис.10.1.2.5. Страница с выгрузкой истории загрузок публикаций в систему.

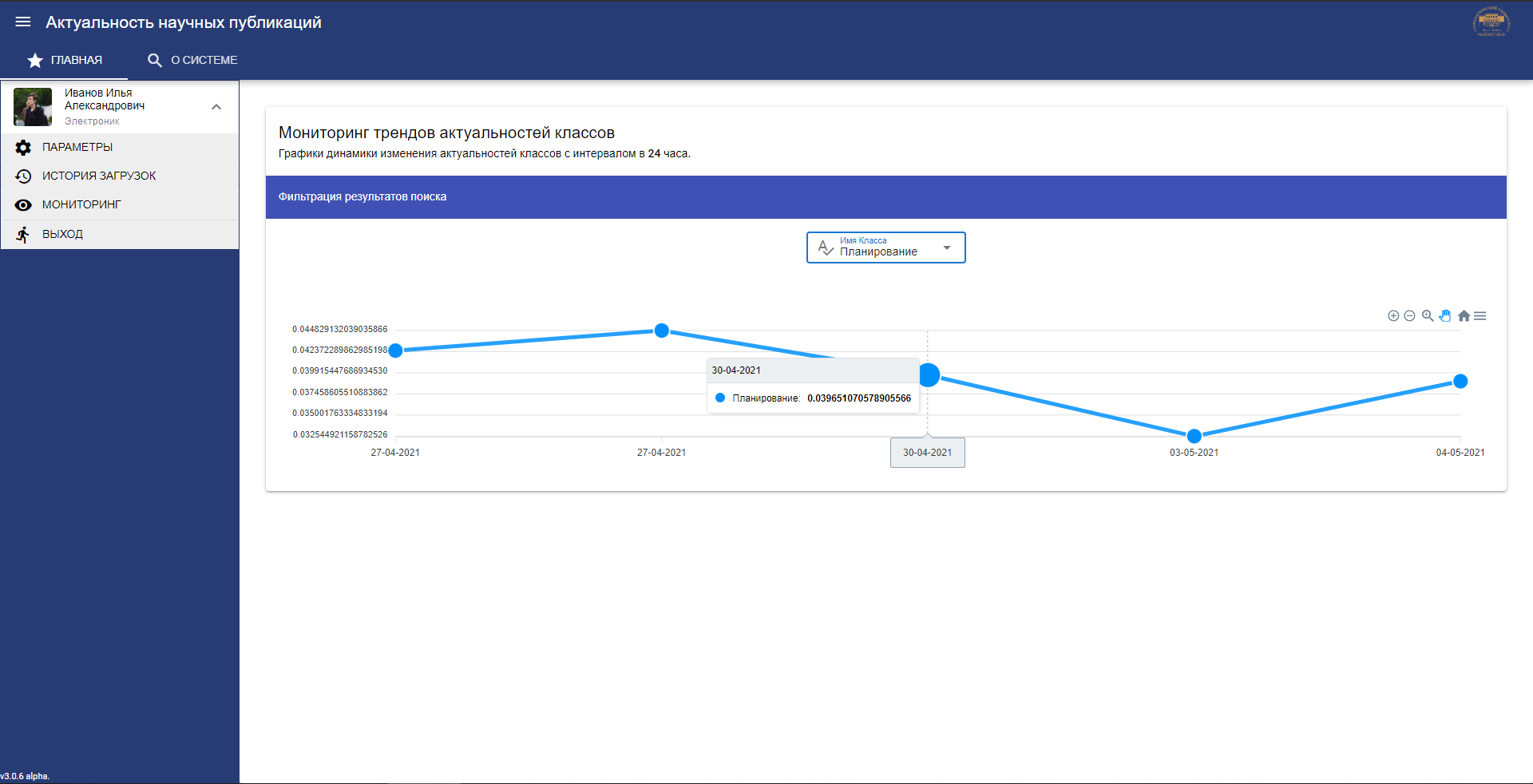


Рис.10.1.2.6. Страница с мониторингом динамики изменения актуальностей классов.

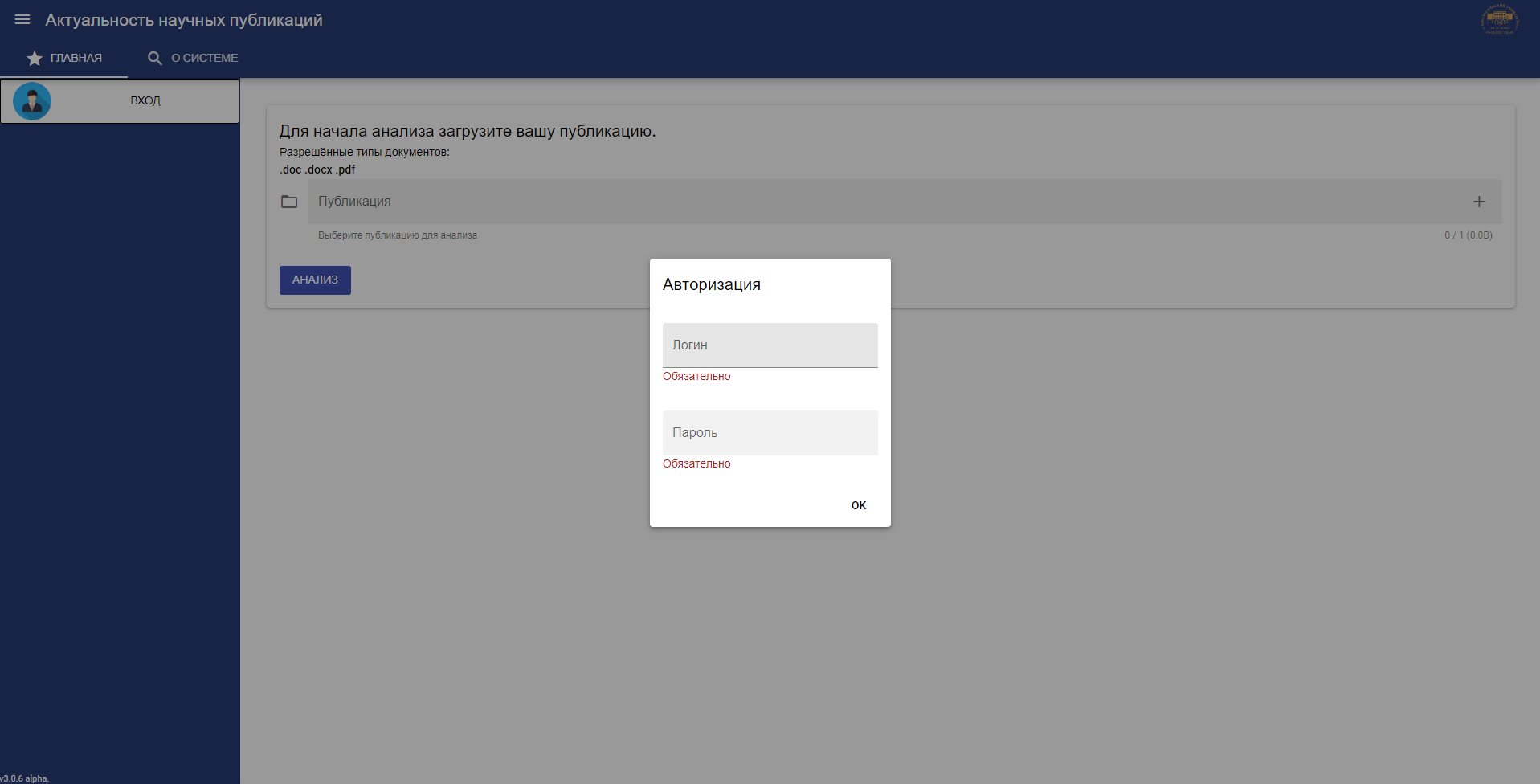


Рис.10.1.2.7 Форма авторизации

# Заключение

В преддипломной практике были выполнены все поставленные задания, включающие:

1. Описание общего алгоритма работы системы;
2. Поиск и рассмотрение существующих рекомендательные систем;
3. Описание базовые терминов и определений в области машинного обучения и рекомендательных систем;
4. Определение преимуществ разрабатываемой системы;
5. Определение архитектуры сети и метрики определения качества обучения;
6. Привести пример результатов разработки;
   1. Алгоритмы (Back-end);
   2. Пользовательский интерфейс (Front-end).

На основе П.7 перечня поставленных задач, сделаны выводы и подобрана оптимальная для решения задачи построения рекомендательной системы архитектура нейронной сети. На основе П.5.2 проанализированы метрики из которых были выбраны те, которые наиболее точно и оптимально помогают оценить качество обучения и наборов данных для него. Определены преимущества разрабатываемой системы, а также примеры разработки серверной и клиентских частей. Построена и обучена нейронная сеть. Определено её предназначение в рамках рассматриваемой задачи. Описаны алгоритмы работы системы и взаимодействия её компонентов.

# Список используемых источников

**Электронные источники**

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. MovieLens Non-commercial, personalized movie recommendations. [Электронный ресурс] // URL: https://movielens.org/ (дата обращения: 03.08.2020)
3. 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).
4. Apache POI. [Электронный ресурс] // URL: https://poi.apache.org (дата обращения: 09.07.2020).
5. OkHTTP. [Электронный ресурс] // URL: https://square.github.io/okhttp (дата обращения: 11.07.2020).
6. FlyWay. [Электронный ресурс] // URL: https://flywaydb.org/documentation (дата обращения: 19.07.2020).
7. IText. [Электронный ресурс] // URL: https://itextpdf.com/ru (дата обращения: 21.07.2020).
8. CoursEra. [Электронный ресурс] // URL: https://www.coursera.org/ (дата обращения: 03.02.2021).
9. MathGarden. [Электронный ресурс] // URL: https://play.google.com/store/apps/details?id=com.nummolt.kinder.garden.math\_serialized&hl=en\_US&gl=US (дата обращения: 20.12.2020).