**Санкт-Петербургский государственный электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**(СПбГЭТУ «ЛЭТИ»)**

|  |  |  |
| --- | --- | --- |
| **Направление** | 09.04.02 Информационные системы и технологии | |
| **Программа** | Управление IT проектами и продуктами | |
| **Факультет** | КТИ | |
| **Кафедра** | АПУ | |
| *К защите допустить* |  | |
| Зав. кафедрой |  | Шестопалов М.Ю. |

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

МАГИСТРА

Тема: Разработка алгоритма семантической декомпозиции текста на ключевые элементы для решения задачи категоризации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студентка |  |  | |  | Терещенко В.Н. |
|  |  | *подпись* | |  |  |
| Руководитель | к.т.н., доцент |  | |  | Кораблев Ю.А. |
|  |  | *подпись* | |  |  |
| Консультанты | к.т.н. |  | |  | Лосева Д.М. |
|  |  | *подпись* | |  |  |
|  | к.т.н. | *По доп разделу* | |  | Заславский М.М. |
|  |  | *подпись* | |  |  |
|  |  | |

Санкт-Петербург

2022**ЗАДАНИЕ**

**на выпускную квалификационную работу**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой АПУ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Шестопалов М.Ю. |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2022 г. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студентка | Терещенко В.Н. | | | | Группа 6371 |
| Тема работы: Разработка алгоритма семантической декомпозиции текста на ключевые элементы для решения задачи категоризации. | | | | | |
| Место выполнения ВКР: Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В.И. Ульянова (Ленина), кафедра АПУ | | | | | |
| Исходные данные (технические требования): OC Windows 10 64-bit, Python 3.8.0 и выше 64-bit. | | | | | |
| Содержание ВКР:  Введение, Обзор предметной области, Формулировка требований к решению, Описание процесса разработки, Исследование, Безопасность жизнедеятельности, Заключение. | | | | | |
| Перечень отчетных материалов: пояснительная записка, иллюстративный материал. | | | | | |
| Дополнительные разделы: Безопасность жизнедеятельности. | | | | | |
|  | | | | | |
| Дата выдачи задания | | | Дата представления ВКР к защите | | |
| «\_01\_» \_\_февраля\_\_\_\_\_2022 г. | | | «\_01\_» \_\_июня\_\_\_\_\_2022 г. | | |
| Студентка | |  | | Терещенко В.Н. | |
| Руководитель к.т.н., доцент | |  | | Кораблев Ю.А. | |

**календарный план выполнения**

**выпускной квалификационной работы**

|  |  |
| --- | --- |
|  | Утверждаю |
|  | Зав. кафедрой АПУ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_ Шестопалов М.Ю. |
|  | «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 г. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студентка | Терещенко В.Н. |  | Группа | 6371 |
| Тема работы: Разработка алгоритма семантической декомпозиции текста на ключевые элементы для решения задачи категоризации. | | | | |

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование работ | Срок выполнения |
| 1 | Обзор литературы по теме работы | 20.01 – 21.02 |
| 2 | Разработка алгоритма | 22.02 – 05.04 |
| 3 | Проведение исследования | 06.04 – 09.04 |
| 4 | Внедрение алгоритма в СППР и тестирование | 06.05 – 15.05 |
| 5 | Оформление пояснительной записки | 10.04 – 22.05 |
| 6 | Оформление иллюстративного материала | 23.05 – 26.05 |
| 7 | Предзащита | 26.05 |

|  |  |  |
| --- | --- | --- |
| Студентка |  | Терещенко В.Н. |
| Руководитель к.т.н., доцент |  | Кораблев Ю.А. |

**РЕФЕРАТ**

Пояснительная записка 59 стр., 8 рис., 3 табл., 32 ист., 2 прил.

Ключевые слова: веб-приложение, выделение ключевых слов, ключевое слово.

Цель работы – исследование современных методов классификации, создание инструмента, автоматически генерирующего оглавление и указатели терминов для массовых открытых онлайн-курсов платформы Stepik.

Объектом разработки является проектируемое веб-приложение.

В работе описано исследование подходов к решению задачи выделения ключевых слов, описан процесс проектирования веб-приложения для создания оглавления и списка терминов. Приведено практическое сравнение статистического и графового подходов в разработанном веб-приложении. При проектировании приложения были использованы язык программирования Python, веб-фреймворк Flask, СУБД PostgreSQL и REST-интерфейс Stepik API. Полученное приложение исследовано: определены время работы алгоритмов выделения ключевых слов Frequency, TopicRank и TextRank, проверены точность алгоритмов методом сравнения с экспертной оценкой.

**ABSTRACT**

The paper contains a study of approaches to solving the problem of highlighting keywords, describes the design process of a web application to create a table of contents and a list of terms. A practical comparison of statistical and graph approaches in the developed web application is given. When designing the application, the programming language Python, the Flask web framework, the PostgreSQL DBMS, and the Stepik API REST interface were used. The resulting application was investigated: the operating time of the Keyword, Frequency, TopicRank, and TextRank keyword extraction algorithms was determined, the accuracy of the algorithms was verified by comparison with an expert assessment.

**содержание**

[**определения, обозначения и сокращения** 8](#_Toc96367284)

[**введение** 9](#_Toc96367285)

[**1.** **Обзор предметной области** 11](#_Toc96367286)

[**1.1.** **Категоризация** 11](#_Toc96367287)

[**1.2.** **Обработка естественного языка** 11](#_Toc96367288)

[**1.3.** **Семантический анализ** 12](#_Toc96367289)

[**1.4.** **Семантическая декомпозиция** 12](#_Toc96367290)

[**1.5.** **Обзор существующих подходов (алгоритмов)** 12](#_Toc96367291)

[**1.6.** **Сравнение существующих подходов (алгоритмов)** 12](#_Toc96367292)

[**1.7.** **Выбор подхода (алгоритма)** 15](#_Toc96367293)

[**1.8.** **Выводы** 16](#_Toc96367294)

[**2.** **Формулировка требований к решению** 17](#_Toc96367295)

[**3.** **Описание процесса разработки** 18](#_Toc96367296)

[**3.1.** **Алгоритм семантической декомпозиции на n-граммы** 18](#_Toc96367297)

[**3.1.1.** **Структура и описание работы алгоритма** 18](#_Toc96367298)

[**3.1.2.** **Технические подробности реализации** 18](#_Toc96367299)

[**3.1.3.** **Сценарий использования** 18](#_Toc96367300)

[**3.1.4.** **Апробация алгоритма** 18](#_Toc96367301)

[**3.2.** **Алгоритм семантической декомпозиции на простые предложения** 18](#_Toc96367302)

[**3.2.1.** **Структура и описание работы алгоритма** 18](#_Toc96367303)

[**3.2.2.** **Технические подробности реализации** 18](#_Toc96367304)

[**3.2.3.** **Сценарий использования** 18](#_Toc96367305)

[**3.2.4.** **Апробация алгоритма** 19](#_Toc96367306)

[**3.3.** **Выводы** 19](#_Toc96367307)

[**4.** **исследование** 20](#_Toc96367308)

[**4.1.** **Исследование времени работы** 20](#_Toc96367309)

[**4.1.1.** **Сценарий эксперимента** 20](#_Toc96367310)

[**4.1.2.** **Методика измерения** 20](#_Toc96367311)

[**4.1.3.** **Используемые программные средства** 20](#_Toc96367312)

[**4.1.4.** **Результаты эксперимента** 21](#_Toc96367313)

[**4.1.5.** **Сравнение и оценка результатов** 21](#_Toc96367314)

[**4.2.** **Исследование точности результатов** 21](#_Toc96367315)

[**4.2.1.** **Сценарий эксперимента** 22](#_Toc96367316)

[**4.2.2.** **Входные данные** 22](#_Toc96367317)

[**4.2.3.** **Результат работы алгоритмов** 22](#_Toc96367318)

[**4.2.4.** **Экспертная оценка** 23](#_Toc96367319)

[**4.2.5.** **Сравнение с экспертной оценкой** 23](#_Toc96367320)

[**4.3.** **Выводы** 24](#_Toc96367321)

[**5.** **безопАсность жизнедеятельности** 25](#_Toc96367322)

[**заключение** 26](#_Toc96367323)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 28](#_Toc96367324)

# **определения, обозначения и сокращения**

В настоящей пояснительной записке применяют следующие термины с соответствующими определениями:

Веб-сайт – совокупность взаимосвязанных веб-страниц, используемых одно доменное имя (адрес).

Веб-страница – документ, обычно в формате html, содержащий какой-либо контент (текст, фото, видео и другое).

Документ – коллекция текста (твит, статья и др.).

Классификатор – алгоритм, или метод, посредством которого проводится классификация.

Классификация – процедура, в ходе которой объекты (предметы, явления) распределяются по группам, или классам, по каким-либо признакам, удобным для их исследования.

Корпус – набор документов схожей тематики.

МООК – массовые открытые онлайн курс (Massive Open Online Course, MOOC).

ОЗУ – оперативное запоминающее устройство.

Реестр – форма систематизации, учета.

СУБД – система управления базами данных.

API – Application Programming Interface (программный интерфейс приложения).

ID – Identifier (идентификатор).

NLP – Natural Language Processing (обработка естественного языка).

REST – Representational State Transfer (передача состояния представления).

SQL – Structured Query Language (язык структурированных запросов).

# **введение**

Кратко (на одну-две страницы) дать оценку современного состояния решаемой научно-технической проблемы, показать актуальность вопроса и необходимость проведения исследования (разработки), указать основную цель работы.

В начале 20 века появился такой метод обучения, как массовые открытые онлайн-курсы (МООК, или (англ.) MOOC – Massive Open Online Course) [1]. Большое количество бесплатных курсов с различными тематиками находятся во всеобщем доступе. [2].

С начала появления МООК и по сегодняшний день данный тип обучения становится все более популярным, так как используются людьми в качестве средства получения как основного образования, так и дополнительного. Однако не во всех курсах теоретический материал дается в удобной для этого форме.

Большое количество массовых онлайн курсов имеет значительные объемы теоретического материала, что вызывает сложности в его усваивании. Отсутствие удобства пользования курсом заключается в потери времени на поиск нужной информации в случае, если слушатель не запомнил точного расположения в курсе нужного термина или примера решения задачи.

Вариантом разрешения данной проблемы может выступать веб-приложение для автоматизированного составления оглавления и списка терминов с их определениями в материалах курса.

**Целью** работы является исследование современных методов классификации и создание инструмента, автоматически генерирующего оглавление и указатели терминов для массовых открытых онлайн-курсов платформы Stepik.

Для достижения цели необходимо выполнить следующие **задачи**:

* сравнить и проанализировать существующие подходы к задаче выделения ключевых слов;
* сформулировать требования к программному продукту;
* разработать программный продукт;
* исследовать свойства программного продукта.

**Объектом** исследования являются материалы онлайн-курсов.

**Предметом** исследования является процесс выделения терминов из материалов курса.

**Практическая значимость** решения заключается в использовании разработанного инструмента слушателями курсов для более удобного использования материалов курса в процессе повторения, обновления уже изученного.

1. **Обзор предметной области**

Для выделения терминов необходим алгоритм их поиска в конкретном тексте. Для этого существуют алгоритмы извлечения ключевых слов.

* 1. **Категоризация**

Существует такое понятие как ключевые слова или словосочетания. Под ключевыми понимают важные слова или словосочетания (фразы), дающие описание содержания текста и позволяющие выявить его тематику [3]. Термины по определению относятся к ключевым словам (словосочетаниям), значит, для их поиска можно применить алгоритмы выделения или извлечения из текстов ключевых слов или словосочетаний (Keyword Extraction).

В данной работе предстоит обрабатывать текстовое содержимое онлайн-курса на естественном языке, что приближает задачу выделения ключевых слов к задаче обработки естественного языка.

* 1. **Обработка естественного языка**

Обработка естественного языка (NLP, Natural language processing) представляет собой область науки, сочетающей в себе математическую лингвистику и машинное обучение. Исходным множеством объектов в NLP является текст или речь на естественном языке. Основными направлениями применения NLP являются распознавание естественного языка, генерация естественного языка и распознавание речи [4].

Для сокращения времени выполнения обработки и увеличения точности результата используются различные процедуры. К таковым относятся:

* перевод всех букв к одному регистру (чаще к нижнему);
* удаление знаков пунктуации, цифр, пробелов;
* токенизация – разбиение текста на более мелкие части (токены);
* использование N-грамм (N-грамма – последовательность из N слов);
* удаление стоп-слов (слов без особой смысловой нагрузки, например, предлоги, местоимения, союзы и другое);
* стемминг – приведение слова к основной форме (нахождение его основы);
* лемматизация – приведение слова к начальной форме (лемма – слово в начальной форме);
* векторизация – отображение текста в векторном пространстве [5].

Различные наборы таких процедур называют предобработкой. Предобработка является первым этапом NLP. Далее следуют сам процесс извлечения ключевых слов.

* 1. **Семантический анализ**

использование нейронных сетей.

* 1. **Семантическая декомпозиция**
  2. **Обзор существующих подходов (алгоритмов)**
  3. **Сравнение существующих подходов (алгоритмов)**

В качестве резюмирования вышеперечисленных характеристик можно составить таблицу (см. табл. 1). В ней наглядно продемонстрированы сходства и различия подходов.

Таблица 1 – Обобщенная характеристика подходов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Критерии | Подходы | | | |
| Статистический | Лингвистический | Графовый | С применением машинного обучения |
| Временные затраты на применение | Время на поиск составляющих корпусов документов (не для всех алгоритмов) | Время на составление лингвистических баз знаний для языков, для которых их еще не существует | Время на составление лингвистических баз знаний для языков, для которых их еще не существует (не для всех алгоритмов) | Время на обучение алгоритмов, время на составление обучающей выборки |
| Зависимость от языка | Не зависит | Зависит | Зависит | Зависит |
| Требование наличия специфических средств для работы | Некоторые алгоритмы требуют корпусы текстов для рассматриваемой предметной области | Алгоритмы требуют наличия лингвистических баз знаний (словарей, грамматик и др.) | Некоторые алгоритмы требуют наличия лингвистических баз знаний на языке обрабатываемого текста | Алгоритмы требуют наличия обучающего корпуса |

Продолжение таблицы 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Особенности подхода | Прост в использовании | Часто применяется для предобработки текста | Требует отсутствия ошибок в тексте | Является самым популярным на сегодняшний день |
| Сфера применения | Языки с бедной морфологией; неспециализированные тексты | Применяется для текстов, лингвистические базы знаний для которых уже разработаны; часто применяется на этапе предобработки текста в сочетании с другими методами | Тексты с классическим типом построения предложений, характерным для рассматриваемого языка | Любые тексты, для которых существуют обучающие корпусы |

Рассматривая лингвистический и графовый подходы, можно заметить, что лингвистический подход может быть использован в сочетании с любым другим подходом (например, для предобработки текста), а многим графовым алгоритмам необходима информация для построения графа. Данные методы отлично сочетаются, что проявляется в хороших результатах таких алгоритмов [3, 8-10]. Примером такого сочетания может быть использование синтаксической или морфологической информации для построения графов в графовых алгоритмах. Это один из видов гибридных подходов.

Алгоритмы статистического подхода и подхода с применением машинного обучения применяют корпусы. Одним корпусы нужны для подсчета уникальности кандидатов в ключевые слова (например, TD-IDF), другим же они нужны для хранения обучающей информации (например, CRF). При этом в одних корпусах содержатся коллекции документов, похожих по теме с обрабатываемым документом, а в других корпусах (обучающих) содержатся размеченные данные.

Алгоритмы лингвистического, графового подходов и подхода с применением машинного обучения имеют различные, часто значительные, затраты по времени на применение, в отличие от алгоритмов статистического подхода, которые, к тому же, являются универсальными для всех языков.

Анализ также показал, что при выборе подходов для извлечения ключевых слов необходимо учитывать предметную область и стиль текста, наличие необходимых лингвистических ресурсов для выбранного языка.

* 1. **Выбор подхода (алгоритма)**

В результате обзора и сравнения походов выделения ключевых слов были выбраны два подхода для интеграции в приложение. В качестве первого подхода был выбран подход, не зависящий от языка текста, – статистический подход. В качестве второго – графовый подход, из-за его более сложной реализации в сравнении со статистическим, а также из-за того, что тексты, содержащиеся в разделах МООК, ввиду своей направленности на образовательную деятельность, не должны содержать орфографических ошибок. В процессе предобработки текстов также были использованы простейшие лингвистические процедуры.

* 1. **Выводы**

В результате обзора предметной области были определены основные термины, необходимые для понимания рассматриваемой темы. При рассмотрении существующих подходов к задаче выделения ключевых слов были определены: общая схема работы, область применения, временные затраты на применение, зависимость от языка и требование наличия специфических средств для работы. Два подхода (статистический и графовый) было решено внедрить в проект для их дальнейшего исследования. Таким образом, было проведено сравнение и анализ существующих подходов выделения ключевых слов.

1. **Формулировка требований к решению**

Исходя из обзора предметной области и анализа проблемы, описанной во введении, принято решение о создании веб-приложения.

Программное решение должно обладать следующими свойствами:

* извлечение терминов из содержимого разделов русскоязычных курсов;
* наличие для каждого найденного термина определения;
* наличие интерфейса для взаимодействия пользователя с приложением;
* наличие интерфейса для взаимодействия сервера приложения с базой данных;
* наличие интерфейса для взаимодействия сервера приложения с платформой Stepik.

1. **Описание процесса разработки**
   1. **Алгоритм семантической декомпозиции на n-граммы**

Перед проектированием веб-приложения необходимо понять, из чего оно должно состоять.

* + 1. **Структура и описание работы алгоритма**

дпдшмлм

* + 1. **Технические подробности реализации**

дпдшмлм

* + 1. **Сценарий использования**

дпдшмлм

* + 1. **Апробация алгоритма**

идпмдлид

* 1. **Алгоритм семантической декомпозиции на простые предложения**
     1. **Структура и описание работы алгоритма**

лалалалал

* + 1. **Технические подробности реализации**

Для реализации веб-приложения был выбран язык Python версии 3.7. Решение использовать Python для реализации решения было принято потому, что данный язык программирования является универсальным языком, применяемым в работе с системами, обрабатывающими HTTP-запросы, с хранилищами баз данных, с шаблонами веб-страниц, с анализом данных и со многим другим [12]. Ниже дается описание основных библиотек и фреймворков, задействованных при проектировании веб-приложения

* + 1. **Сценарий использования**

После запуска веб-приложения работа с ним может проводиться по следующему сценарию:

1. На начальной странице пользователь вводит ID курса.
2. Веб-приложение загружает список разделов курса в порядке, в котором они находятся в МООК.
3. Пользователь выбирает интересующий раздел, кликая по нему.
4. На экране появляются три группы по 10 терминов из выбранного раздела в порядке от слова с самой большой вероятностью того, что оно является термином в данном разделе, к слову с наименьшим значением вероятности. Во главе каждой группы терминов выводится название метода, которым было проведено их выделение.
5. Пользователь кликает на термин.
6. Под выбранным термином появляется его определение.
   * 1. **Апробация алгоритма**

Здесь о том, что разработанный алгоритм до настоящего времени не внедрялся в программные продукты организации, но в будущем может быть использован в СППР для службы технической поддержки ООО ГП.

Пример: «Разработанное решение до настоящего момента не внедрялось в организации, но было опубликовано на Docker Hub. Docker Hub – это общедоступный реестр, поддерживаемый Docker, хранящий большую коллекцию Docker-образов».

* 1. **Выводы**

Результатом разработки является готовое веб-приложение для генерации оглавления и списка терминов для МООК, в котором были реализованы алгоритмы выделения ключевых слов Frequency, TextRank и TopicRank. С помощью данных алгоритмов приложение проводит анализ русскоязычных курсов Stepik, в результате которого из содержимого разделов МООК извлекаются термины. К терминам подбираются определения из Википедии. Приложение интегрировано со Stepik API и с PostgreSQL и имеет пользовательский интерфейс.

1. **исследование**
   1. **Исследование времени работы**

Было проведено исследование времени работы алгоритмов Frequency, TextRank и Topic Rank.

* + 1. **Сценарий эксперимента**

Измеряемой величиной является время, затраченное на работу каждого из алгоритмов. Единица измерения – секунды.

Измерение проводилось путем фиксирования времени начала выполнения алгоритма и окончания его выполнения, затем найденная разность данных значений приравнивалась ко искомому значению затраченного времени.

Процедура проводилась 100 раз для каждого эксперимента.

* + 1. **Методика измерения**

Измерение происходит машине со следующими характеристиками:

* операционная система – Windows 10 Pro;
* системные характеристики:
  + процессор – Intel Core i3-7100U CPU 2.4GHz (2 ядра);
  + ОЗУ – 8Гб;
  + тип системы – х64.

Эксперимент проводился для текстового содержимого раздела «Функции. Словари. Интерпретатор. Файлы. Модули» курса «Программирование на Python» (ID 67).

* + 1. **Используемые программные средства**

Эксперимент проводился с использованием средств Python, а именно с помощью библиотеки time и ее метода monotonic() [32].

* + 1. **Результаты эксперимента**

Было рассчитано минимальное, максимальное и среднее время выполнения каждого алгоритма. Полученные данные, приведенные в секундах с точностью до тысячных, представлены в табл. 2.

Таблица 2 – Обработанные данные (для алгоритмов)

|  |  |  |  |
| --- | --- | --- | --- |
| Данные | Алгоритмы | | |
| Frequency | TopicRank | TextRank |
| Минимальное время работы | 8,032 | 5,892 | 7,504 |
| Среднее время работы | 8,761 | 6,556 | 8,165 |
| Максимальное время работы | 13,100 | 11,014 | 12,563 |

Таблица с необработанными данными представлена в прил. Б.

* + 1. **Сравнение и оценка результатов**

Как видно из расчетов, разность между максимальным и средним временем работы гораздо больше, чем разность между средним и минимальным временем, что говорит о том, что итерации, показавшие сравнимое с максимальным время, достигалось малое количество раз. Данную гипотезу можно проверить, ознакомившись с таблицей в приложении Б. Подобное поведение может быть связано с внешними факторами, такими как скорость интернет-соединения, время ответа сервера Stepik.

Сравнивая среднее время работы алгоритмов, видно, что лучшее время работы показывает TopicRank, алгоритм TextRank оказывается на втором месте, а Frequency на третьем. Все три алгоритма реализованы с применением различных готовых решений, имеющихся в библиотеках обработки текстов, что объясняет разную скорость работы алгоритмов. Работа методов таких библиотек не отличается быстродействием, что объясняет невысокую скорость выполнения алгоритмов.

* 1. **Исследование точности результатов**

В рассмотрении задач выделения ключевых слов большую роль играет точность их выделения.

* + 1. **Сценарий эксперимента**

В ходе эксперимента запускается веб-приложение, результатом чего становится получение трех списков с терминами. Далее слова в каждом из списков сравниваются со словами эталонного списка, состоящего из такого же количества ключевых слов, таким образом выделяется точность алгоритма в процентах.

Под эталонным списком понимается список отобранных вручную ключевых слов из того же документа.

В списках терминов и в эталонном списке содержится по 10 слов.

* + 1. **Входные данные**

Эксперимент проводится на базе текстового содержимого раздела «Введение» курса «Как писать научные статьи» (ID 10524).

* + 1. **Результат работы алгоритмов**

Результатом работы алгоритмов стали следующие ключевые слова:

1. Frequency:
   * научный;
   * статья;
   * работа;
   * курс;
   * комментарий;
   * исследование;
   * результат;
   * источник;
   * текст;
   * пример;
2. TopicRank:
   * научный;
   * работа;
   * курс;
   * комментарий;
   * исследование;
   * результат;
   * источник;
   * текст;
   * список;
   * пример;
3. TextRank:
   * метод;
   * явно;
   * вывод;
   * введение;
   * аннотация;
   * слово;
   * state;
   * анализ;
   * название;
   * совет.
     1. **Экспертная оценка**

Эталонный список содержит следующие ключевые слова:

* научный;
* статья;
* исследование;
* обоснованность;
* анализ;
* аннотация;
* курс;
* автор;
* публикация;
* источник.

Словосочетание «научная статья» было разделено ввиду того, что алгоритмы ведут поиск юниграмм – N-грамм, состоящих из одного слова.

* + 1. **Сравнение с экспертной оценкой**

В ходе сравнения было вычислено отношение числа совпавших слов к общему количеству слов в списке. Результаты сравнения представлены в табл. 3.

Таблица 3 – Сравнение с экспертной оценкой

|  |  |  |  |
| --- | --- | --- | --- |
| Данные | Алгоритмы | | |
| Frequency | TopicRank | TextRank |
| Точность, % | 50 | 50 | 20 |

Согласно таблице 3, алгоритм TextRank не применим к задаче выделения ключевых слов, так как его точность составляет всего 20%. Это можно объяснить тем, что данный алгоритм основан на алгоритме PageRank, для построения графа которого используются связи между элементами. В данной же реализации алгоритма на вход подается список уже отделенных друг от друга слов, что мешает работе алгоритма.

В связи с этим же фактом, алгоритм TopicRank, показавший средние результаты (50%), может улучшить свой результат при иной реализации этапа предобработки.

Алгоритм Frequency, показавший также средний результат (50%), тоже можно улучшить с помощью использования N-грамм. Однако, даже в этом случае, он с малой вероятностью покажет стопроцентный результат, так как не предназначен для специализированных текстов.

* 1. **Выводы**

Таким образом, было проведено исследование свойств программного продукта, таких как время работы алгоритмов, выполняющих выделение терминов из текста МООК, и их точность. В ходе исследования было выявлено, что алгоритм TopicRank быстрее всех справляется с задачей (6,556 с.) и имеет среднюю точность (50%); алгоритм TextRank на втором месте по времени (8,165 с.), но на последнем по точности (20%); Frequency на третьем месте по времени (8,761 с.) и имеет такую же точность, как и TopicRank – 50%.

1. **безопАсность жизнедеятельности**

В данном разделе проверяется соответствие пользовательского интерфейса выполненного веб-приложения стандартам взаимодействия пользователь-система, принятым согласно требованиям группы ГОСТ 9241. Рассмотрены удобство и интуитивность использования пользователем предложенных функций, а также удобство чтения выведенного материала, запрашиваемого пользователем.

В работе наиболее важной частью анализа качества взаимодействия пользователя с веб-приложением является оценка удобства пользования интерфейсом, по причине того, что это единственная форма взаимодействия между приложением и пользователем. Оценка интерфейса проводится по визуальным качествам и по количеству операций необходимых пользователю для получения ожидаемого результата.

# **заключение**

В результате работы над выпускной квалификационной работой были изучены походы к задаче выделения ключевых слов, выполнено проектирование программного продукта, представляющее собой веб-приложение, автоматически генерирующее оглавление и указатели терминов для массовых открытых онлайн-курсов платформы Stepik.

В процессе обзора предметной области были определены основные термины рассматриваемой области, дано описание подходов выделения ключевых слов, проанализированы области их применения, временные затраты на применение, наличие требований использования специфических средств для работы и особенности подходов. Проведен выбор двух подходов, реализуемых в дальнейшем в веб-приложении. Выбранными подходами стали статистический и графовый.

На основании обзора предметной области и анализа проблемы, описанной во введении, были выдвинуты требования к разрабатываемому решению. Учитывая данные требования, была достигнута основная задача работы – спроектирован программный продукт для автоматического генерирования оглавления и указателей терминов для русскоязычных массовых открытых онлайн-курсов. Разработанное решение представлено веб-приложением на языке программирования Python, с использованием веб-фреймворка Flask и СУБД PostgreSQL. Полученное программное решение имеет пользовательский интерфейс. Приложение интегрируется платформой Stepik с помощью REST-интерфейса Stepik API, а с хранилищем данных с помощью библиотеки **SQLAlchemy**. Посредством анализа алгоритмами Frequency, TopicRank и TextRank из выбранного раздела МООК выделяются термины, для которых имеются определения.

Было проведено исследование свойств программного решения: исследование времени работы алгоритмов выделения ключевых слов и их точности. По результатам исследования быстродействия алгоритмов самым быстрым оказался алгоритм TopicRank (6,556 с.), а самым медленным – Frequency (8,761 с.). Это объясняется использованием готовых решений для обработки текста, что сильно влияет на скорость работы алгоритмов. В результате исследования точности, ни один алгоритм не показал отличных результатов из-за неподходящего для них сценария реализации этапа предобработки (TopicRank, 50%; TextRank, 20%) или из-за неподходящего проверяемому алгоритму типа текста (Frequency, 50%).

В будущем данную проблем, а также проблему малой скорости работы планируется решить для алгоритмов, потенциально подходящих для задачи выделения терминов из содержимого МООК, то есть для графовых алгоритмов TopicRank и TextRank.

Кратко (на одну – две страницы) описать основные результаты работы, проанализировать их соответствие поставленной цели работы, показать рекомендации по конкретному использованию результатов исследования и перспективы дальнейшего развития работы.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Что такое МООС (Массовые открытые онлайн курсы) [Электронный ресурс]. URL: https://etu.ru/ru/on-line-obuchenie/mooc (дата обращения: 15.05.2020).
2. Чекалина Т. А., Тумандеева Т. В., Максименко Н. В. / Основные направления и перспективы развития онлайн-обучения // Профессиональное образование в России и за рубежом. – 2018. – (№) 3 (31). – С. 44-52. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/osnovnye-napravleniya-i-perspektivy-razvitiya-onlayn-obucheniya/viewer (дата обращения: 15.05.2020).
3. Ванюшкин А. С., Гращенко Л. А. / Методы и алгоритмы извлечения ключевых слов // Новые информационные технологии в автоматизированных системах. – 2016. – (№) 19. – С. 85-93. [Электронный ресурс]. URL: https://cyberleninka.ru/article/n/metody-i-algoritmy-izvlecheniya-klyuchevyh-slov/viewer (дата обращения: 24.05.2020).
4. Manning C. D., Manning C. D., Schütze H. Foundations of statistical natural language processing. – MIT press, 1999. 680p.
5. Лазарева О. Ю., Боломутова М. С. / Методы выделения ключевых слов в контексте электронных обучающих систем // Молодой ученый. – 2016. – (№) 26 (130). С. 143-146. [Электронный ресурс] URL: https://moluch.ru/archive/130/35952/ (дата обращения: 16.05.2020).
6. Лекция. Задачи Data Mining. Классификация и кластеризация // НОУ ИНТУИТ. [Электронный ресурс] URL: https://www.intuit.ru/studies/courses/6/6/lecture/166 (дата обращения: 16.05.2020).
7. Keyword Extraction: A Guide to Finding Keywords in Text // MonkeyLearn. [Электронный ресурс] URL: https://monkeylearn.com/keyword-extraction/ (дата обращения: 17.05.2020).
8. Girish K.P. Keyword Extraction from a Single Document Using Centrality Measures / Pattern Recognition and Machine Intelligence, Kolkata, India, December 18–22, 2007 / Kolkata, 2007, pp. 503–510.
9. Litvak M. Graph-based Keyword Extraction for Single-Document Summarization / Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, Manchester, United Kingdom, August 2008 / Manchester, 2008, pp. 17–24.
10. Григорьева Е. Г., Клячин В. А., Помельников Ю. В., Попов В. В. / Алгоритм выделения ключевых слов на основе графовой модели лингвистического корпуса // Вестник ВолГУ. Серия 2, Языкознание. – 2017. – Т. 16 № 2. – С. 58-67. [Электронный ресурс] URL: https://cyberleninka.ru/article/n/algoritm-vydeleniya-klyuchevyh-slov-na-osnove-grafovoy-modeli-lingvisticheskogo-korpusa/viewer (дата обращения: 24.05.2020).
11. Лекция. Принципы работы и структура Web-приложений на основе ASP.NET // НОУ ИНТУИТ [Электронный ресурс] URL: https://www.intuit.ru/studies/courses/1139/250/lecture/6422 (дата обращения: 19.05.2020).
12. Welcome to Python.org // Python [Электронный ресурс] URL: https://www.python.org/ (дата обращения: 20.05.2020).
13. О нас // Stepik [Электронный ресурс] URL: https://welcome.stepik.org/ru/about (дата обращения: 20.05.2020).
14. StepikOrg/Stepik-API: API documentation and Examples // GitHub [Электронный ресурс] URL: https://github.com/StepicOrg/Stepik-API (дата обращения: 20.05.2020).
15. Welcome to Flask // Flask Documentation (1.1.x) [Электронный ресурс] URL: http://flask.pocoo.org/ (дата обращения: 20.05.2020).
16. SQLAlchemy – The Database Toolkit for Python // SQLAlchemy [Электронный ресурс] URL: https://www.sqlalchemy.org/ (дата обращения: 20.05.2020).
17. PostgreSQL: The world’s most advanced open source database // PostgreSQL [Электронный ресурс] URL: https://www.postgresql.org/ (дата обращения: 20.05.2020).
18. Collections – Container datatypes // Python 3.8.3 documentation [Электронный ресурс] URL: https://docs.python.org/3/library/collections.html (дата обращения: 21.05.2020).
19. Boudinfl/pke: Python Keyphrase Extraction module // GitHub [Электронный ресурс] URL: https://github.com/boudinfl/pke (дата обращения: 21.05.2020).
20. Pke documentation // pke 1.8 documentation [Электронный ресурс] URL: https://boudinfl.github.io/pke/build/html/index.html (дата обращения: 23.05.2020).
21. Docker Documentation // Docker Documentation. [Электронный ресурс] URL: https://docs.docker.com/ (дата обращения: 20.05.2020).
22. Заглавная страница // Википедия – свободная энциклопедия. [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Заглавная\_страница (23.05.2020).
23. Русская Википедия // Википедия. [Электронный ресурс] URL: https://ru.wikipedia.org/wiki/Русская\_Википедия (дата обращения: 23.05.2020).
24. Bougouin A. TopicRank: Graph-Based Topic Ranking for Keyphrase Extraction / International Joint Conference on Natural Language Processing, Nagoya, Japan, 14-18 October 2013 / Nagoya, 2013, pp 543–551.
25. Unsupervised models (TopicRank) // pke 1.8 documentation [Электронный ресурс] URL: https://boudinfl.github.io/pke/build/html/unsupervised.html#topicrank (дата обращения: 23.05.2020).
26. Mihalcea R. TextRank: Bringing Order into Texts / EMNLP, July 2004 / EMNLP, July 2004, № 32.
27. Unsupervised models (TextRank) // pke 1.8 documentation [Электронный ресурс] URL: https://boudinfl.github.io/pke/build/html/unsupervised.html#textrank (дата обращения: 23.05.2020).
28. Алгоритм PageRank // Data Science [Электронный ресурс] URL: http://datascientist.one/pagerank-algorithm/ (дата обращения: 23.05.2020).
29. Docker Hub // Docker Hub [Электронный ресурс] URL: https://hub.docker.com/repository/docker/vika2108991/vkr (дата обращения 24.05.2020).
30. Moevm/bsc\_tereshchenko // GitHub [Электронный ресурс] URL: https://github.com/moevm/bsc\_tereshchenko (дата обращения: 24.05.2020).
31. Разработка модели алгоритма составления оглавления и указателя терминов для массовых открытых онлайн курсов. / Терещенко В. Н., Заславский М. М. // 73-я научно-техническая конференция профессорско-преподавательского состава СПбГЭТУ «ЛЭТИ», СПб, 5 февраля 2020 г. / СПбГЭТУ «ЛЭТИ», СПб, 2020.
32. 15.3. time – Time access and conversions // Python 2.7.18 documentation. [Электронный ресурс] URL: https://docs.python.org/2/library/time.html (дата обращения: 24.05.2020).