**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: Коммерциализация результатов исследований в рамках НИР. | | | | | |
|  | | | | | |
| 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: ключевое слово, ключевое словосочетание, ключевая фраза, ключевое предложение, семантическая декомпозиция, классификация, категоризация.

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](#_Toc96536728)

[**введение** 9](#_Toc96536729)

[**1.** **Обзор предметной области** 11](#_Toc96536730)

[**1.1.** **Классификация и категоризация текстов на естественном языке** 11](#_Toc96536731)

[**1.2.** **Анализ основных методов (подходов) обработки текстов на естественном языке** 11](#_Toc96536732)

[**1.3.** **Семантическая декомпозиция как один из инструментов семантического анализа текстов** 12](#_Toc96536733)

[**1.4.** **Сравнительный анализ существующих алгоритмов семантической декомпозиции и выбор базовых алгоритмов для реализации** 12](#_Toc96536734)

[**1.5.** **Выводы** 16](#_Toc96536735)

[**2.** **ПОСТАНОВКА ЗАДАЧИ** 17](#_Toc96536736)

[**3.** **Описание Теоретических основ разрабатываемых алгоритмов** 18](#_Toc96536737)

[**3.1.** **Алгоритм семантической декомпозиции на n-граммы** 18](#_Toc96536738)

[**3.1.1.** **Математические основы алгоритма** 18](#_Toc96536739)

[**3.1.2.** **Графическое описание алгоритма** 18](#_Toc96536740)

[**3.1.3.** **Сценарий применения алгоритма для решения задачи категоризации** 18](#_Toc96536741)

[**3.2.** **Алгоритм семантической декомпозиции на предложения** 18](#_Toc96536742)

[**3.2.1.** **Математические основы алгоритма** 18](#_Toc96536743)

[**3.2.2.** **Графическое описание алгоритма** 18](#_Toc96536744)

[**3.2.3.** **Сценарий применения алгоритма для решения задачи категоризации** 18](#_Toc96536745)

[**3.3.** **Выводы** 18](#_Toc96536746)

[**4.** **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ** 19](#_Toc96536747)

[**4.1.** **Алгоритм семантической декомпозиции на n-граммы** 19](#_Toc96536748)

[**4.1.1.** **Технические подробности программной реализации алгоритма** 19](#_Toc96536749)

[**4.1.2.** **Тестирование алгоритма** 19](#_Toc96536750)

[**4.1.3.** **Сценарий использования** 19](#_Toc96536751)

[**4.1.4.** **Применение алгоритма для решения задачи категоризации** 19](#_Toc96536752)

[**4.2.** **Алгоритм семантической декомпозиции на предложения** 19](#_Toc96536753)

[**4.2.1.** **Технические подробности программной реализации алгоритма** 19](#_Toc96536754)

[**4.2.2.** **Тестирование алгоритма** 19](#_Toc96536755)

[**4.2.3.** **Сценарий использования** 19](#_Toc96536756)

[**4.2.4.** **Применение алгоритма для решения задачи категоризации** 20](#_Toc96536757)

[**4.3.** **Выводы** 20](#_Toc96536758)

[**5.** **исследование** 21](#_Toc96536759)

[**5.1.** **Исследование времени работы** 21](#_Toc96536760)

[**5.1.1.** **Сценарий эксперимента** 21](#_Toc96536761)

[**5.1.2.** **Методика измерения** 21](#_Toc96536762)

[**5.1.3.** **Используемые программные средства** 21](#_Toc96536763)

[**5.1.4.** **Результаты эксперимента** 22](#_Toc96536764)

[**5.1.5.** **Сравнение и оценка результатов** 22](#_Toc96536765)

[**5.2.** **Исследование точности результатов** 22](#_Toc96536766)

[**5.2.1.** **Сценарий эксперимента** 23](#_Toc96536767)

[**5.2.2.** **Входные данные** 23](#_Toc96536768)

[**5.2.3.** **Результат работы алгоритмов** 23](#_Toc96536769)

[**5.2.4.** **Экспертная оценка** 24](#_Toc96536770)

[**5.2.5.** **Сравнение с экспертной оценкой** 24](#_Toc96536771)

[**5.3.** **Выводы** 25](#_Toc96536772)

[**6.** **Коммерциализация результатов исследований в рамках НИР** 26](#_Toc96536773)

[**заключение** 27](#_Toc96536774)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 30](#_Toc96536775)

# **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**

В настоящее время системы обработки естественного языка используются повсеместно, начиная от переводчика и поисковой системы в браузере, заканчивая умными чат-ботами и голосовыми помощниками. Такие системы внедряются в совершенно различные структуры и упрощают их использование.

Недавно для компании ООО «Газпром переработка» была разработана система поддержки принятия решений для автоматизации работы службы технической поддержки. Данная СППР автоматически обрабатывает запросы, поступающие на 1-ую линию технической поддержки. Обработка главным образом заключается в категоризации запросов и направлении их по различным отделам компании.

Данная тема магистерской диссертации была выбрана с целью исследования влияния различных вариантов разделения текста запроса на результат работы СППР, а именно на результат классификации и, соответственно, категоризации текстовых запросов.

В большинстве случаев наличие среди ключевых слов устойчивых словосочетаний ведет положительно влияет на вероятность правильного категорирования специализированных текстов. **Цель** данной работы – проверить эту гипотезу для текстов обращений, поступающих в техническую поддержку компании Газпром переработка, и предложить вариант доработки проекта с учетов результата исследования.

Для достижения цели были поставлены следующие **задачи**:

* сравнить и проанализировать существующие подходы семантической декомпозиции текста;
* разработать алгоритм и протестировать его (это тоже этап разработки);
* исследовать качество разработанного алгоритма;
* провести внедрение алгоритма в проект (в случае успешного исследования).

**Объектом** исследования являются тексты запросов, поступившие в службу поддержки.

**Предметом** исследования является процесс семантической декомпозиции текста запроса на ключевые элементы.

**Практическая значимость** решения заключается в использовании разработанного алгоритма для улучшения результатов категоризации обращений и, соответственно, для оптимизации работы СППР для службы технической поддержки компании Газпром переработка.

1. **Обзор предметной области**

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

Разрабатываемый алгоритм является частью системы, предназначенной для категоризации запросов.

* 1. **Классификация и категоризация текстов на естественном языке**

Классификаций или категоризацией называют процесс отнесения входных текстовых документов к одной из групп данных (классу или категории).

Эти две интеллектуальные задачи следует отличать друг от друга. Термин *класс* можно определить, как совокупность предметов или понятий (образов), выделенных и сгруппированных по определенным признакам или правилам. Под *классификацией* понимается отнесение некоторого документа к классу, выполняемое по формальным правилам, исходя из совокупности признаков. Границы классов определены достаточно точно: документ относится к данному классу, если известно, что он обладает необходимым числом признаков, характерных для этого класса.

*Категория* же определяет лишь некоторые общие свойства образов и связи между ними. Задача *категоризации*, т. е. определения отношения данного документа к некоторой категории, гораздо менее определена, чем задача отношения к классу. Границы различных категорий являются нечеткими, расплывчатыми, и обычно сама категория понимается не через формальное определение, а только в сравнении с другими категориями.

Таким образом, задачей систем-классификаторов является установление принадлежности образа к одному из формально определенных классов. Примерами такой задачи является задача классификации растений в ботанике, классификация химических веществ по их свойствам и типам возможных реакций, в которые они вступают, и другие. Формальные признаки могут быть определены посредством правил типа “если-то”, а системы, оперирующие с такими правилами, получили название *экспертных систем*. Традиционной областью применения классификаторов на нейронных сетях является экспериментальная физика высоких энергий, где одной из актуальных задач выступает выделение среди множества зарегистрированных в эксперименте событий с элементарными частицами событий, представляющих интерес для данного эксперимента.

Таким образом, проблема категоризации является более сложной в сравнении с проблемой классификации. В категоризации помимо отнесения документа к какой-либо группе, требуется определить сами эти группы, т. е. сформировать категории.

В случае обучения с учителем формирование категорий происходит методом проб и ошибок на основе примеров с известными ответами, предоставляемыми экспертом. Эксперта обычно называют “супервизором” или учителем. Учитель управляет обучением при помощи изменения параметров связей и, реже, самой топологии сети.

Задачей системы-категоризатора является формирование обобщающих признаков в совокупности примеров. При увеличении числа примеров несущественные, случайные признаки сглаживаются, а часто встречающиеся – усиливаются, при этом происходит постепенное уточнение границ категорий. Хорошо обученная нейросетевая система способна извлекать признаки из новых примеров, ранее неизвестных системе, и принимать на их основе приемлемые решения.

Важно отметить различие в характере неявных “знаний”, запомненных искусственной нейронной сетью, и явных, формальных “знаний”, заложенных в экспертных системах. Некоторые сходства и различия представлены в следующей таблице.

|  |  |  |
| --- | --- | --- |
| Критерий | **Экспертные системы (ЭС)** | **Нейросетевые системы (НС)** |
| Источник знаний | Формализованный опыт эксперта, выраженный в виде логических утверждений (правил и фактов), безусловно принимаемых системой | Совокупный опыт эксперта-учителя, отбирающего примеры для обучения, и индивидуальный опыт обучающейся на этих примерах нейронной сети |
| Характер знаний | Формально-логическое “левополушарное” знание в виде правил | Ассоциативное “правополушарное” знание в виде связей между нейронами сети |
| Развитие знаний | В форме расширения совокупности правил и фактов (базы знаний) | В форме дообучения на дополнительной последовательности примеров, с уточнением границ категорий и формированием новых категорий |
| Роль эксперта | Задает на основе правил полный объем знаний экспертной системы | Отбирает характерные примеры, не формулируя специально обоснование своего выбора |
| Роль искусственной системы | Поиск цепочки фактов и правил для доказательства суждения | Формирование индивидуального опыта в форме категорий, получаемых на основе примеров, и категоризация образов |

Различия в характере экспертных и нейросетевых систем обуславливают и различия в их сферах применения. Экспертные системы применяются в узких предметных областях с хорошо структурированными знаниями, например в классификации неисправностей конкретного типа оборудования, фармакологии, анализе химсостава проб и т. д. Нейронные сети применяются кроме перечисленных областей и в задачах с плохо структурированной информацией, например при распознавании образов, рукописного текста, анализе речи и т. д. (<https://pandia.ru/text/79/574/8277.php#:~:text=Классификация%20и%20категоризация.%20В%20случае%2C,выполняет%20классификацию%20или%20категоризацию%20данных>)

* + 1. **Этапы процесса категоризации текстов**

Решение задач категоризации и классификации состоят из четырех последовательных этапов:

* предобработка и индексация документов;
* уменьшение размерности пространства признаков;
* построение и обучение классификатора с помощью методов машинного обучения;
* оценка качества классификации.

На рисунке 1 представлена общая схема процесса категоризации.



Рисунок 1 – Этапы процесса категоризации текстов (скрин из статьи, можно перерисовать)

Рассмотрим этапы, предшествующие работе классификатора, так как алгоритм семантической декомпозиции будет встроен именно в эту часть системы.

Предварительная обработка текста включает нормализацию (перевод всех букв к одному регистру (чаще к нижнему), удаление знаков пунктуации, цифр, пробелов), токенизацию (разбиение текста на более мелкие части – токены), удаление стоп-слов (семантически нейтральных слов, таких как союзы, предлоги, артикли и пр.), морфологический анализ (разметка по частям речи; опционально), лемматизация и стемминг. Это позволяет значительно сократить размерность пространства. В результате в качестве признаков документа выступают все значимые слова, встречающиеся в документе.

Немного о процедурах предварительной обработки.

* **Нормализация и токенизация**. В самом начале обработки текст приводится к единообразному виду (единый регистр слов, отсутствие знаков пунктуации, расшифрованные сокращения, словесное написание чисел и т.д.), что необходимо для применения унифицированных методов обработки текста, и разделяется на более мелкие единицы (на символы, слова или предложения) для того, чтобы иметь возможность провести последующие процедуры.
* **Лемматизация и стемминг**. При построении классификатора текстов нет смысла различать формы одного и того же слова. Это приводит к неоправданному разрастанию словаря, дроблению статистики, увеличению ресурсоёмкости и снижению качества модели.
  + Лемматизация – приведение каждого слова в документе к его нормальной форме (в русском языке: для существительных – именительный падеж, единственное число; для прилагательных – именительный падеж, единственное число, мужской род; для глаголов, причастий, деепричастий – глагол в инфинитиве). Существуют специальные программы – лемматизаторы, обычно основанные на явном хранении грамматического словаря со всеми формами слов. Недостатком лемматизации является трудоёмкость составления словарей, и, как следствие, их неполнота, особенно по части специальной терминологии и неологизмов, которые как раз и представляют наибольший интерес для тематического моделирования.
  + Стемминг – отбрасывание изменяемых частей слов (главным образом, окончаний). Она не требует хранения словаря всех слов и основана на правилах морфологии языка. Недостатком стемминга является большее число ошибок. Стемминг лучше подходит для английского языка, но хуже для русского.
* **Отбрасывание стоп-слов**. Слова, встречающиеся во многих текстах различной тематики, бесполезны при классификации, и могут быть отброшены. К ним относятся предлоги, союзы, числительные, местоимения, некоторые глаголы, прилагательные и наречия. Число таких слов обычно варьируется в пределах нескольких сотен. Их отбрасывание почти не влияет на длину словаря, но может приводить к заметному сокращению длины некоторых текстов.
* **Отбрасывание редких слов**. Слова, встречающиеся в длинном тексте слишком редко, также можно отбрасывать, полагая, что данное слово не имеет принципиального значения в данном тексте.
* **Выделение ключевых фраз**. При обработке коллекций научных, юридических или других специальных текстов вместо отдельных слов выделяют ключевые фразы — словосочетания, являющиеся устойчивыми оборотами или терминами в данной предметной области. Это отдельная и довольно сложная задача, которая может решаться методами машинного обучения с привлечением экспертов для формирования обучающих выборок и контроля качества автоматического выделения терминов [1].

(http://www.machinelearning.ru/wiki/images/0/02/Romanenko2012bach.pdf)

**Индексация** документов представляет собой построение некоторой числовой модели текста, которая переводит текст в удобное для дальнейшей обработки представление.

Распространенные модели индексации:

* модель «мешка слов» (bag-of-words) позволяет представить документ в виде многомерного вектора слов и их весов в документе [2]. Другими словами, каждый документ – это вектор в многомерном пространстве, координаты которого соответствуют номерам слов, а значения координат – значениям весов;
* модель индексации Word2vec [3] представляет каждое слово в виде вектора, который содержит информацию о контекстных (сопутствующих) словах;
* модель индексации, основанная на учете n-грамм [2], то есть последовательностей из соседних токенов.

(https://cyberleninka.ru/article/n/metody-avtomaticheskoy-klassifikatsii-tekstov/viewer)

Далее следует этап выбора или извлечения признаков.

Извлечение признаков в методе текстового представления модели векторного пространства соответствует выбору элементов признаков и вычислению весов признаков. Основная идея выбора функций состоит в том, чтобы независимо ранжировать исходные элементы (термины) в соответствии с определенным индексом оценки, выбирать некоторые элементы функций с наивысшими баллами и отфильтровывать оставшиеся элементы функций. Обычно используемые оценки включают частоту документов, взаимную информацию, получение информации, статистику χ² и т. Д. Чтобы  
Вес признака - это в основном классический метод TF-IDF и метод его расширения. Основная идея состоит в том, что важность слова прямо пропорциональна частоте появления слова в категории и обратно пропорциональна количеству вхождений всех категорий. . Чтобы (https://russianblogs.com/article/52941084692/)

Далее сказать, что дальше идет сам классификатор, а это огромная тема, которая может быть подробно рассмотрена при дальнейшем исследовании, поэтому на ней мы не будем останавливаться.

* + 1. **Ключевые элементы**

Далее можно вставить главу про ключевые элементы (КС и расширение этого понятия с помощью n-грамм).

И потом перейти в главу методов построения n-грамм.

Под ключевыми элементами в данной работе понимаются токены или n-граммы, которые имеют наибольшие показатели весов признаков. Ключевыми элементами могут являться ключевые слова (термины), словосочетания или фразы.

ЗДЕСЬ НУЖНО СКАЗАТЬ О ТОМ, ЧТО ИСПОЛЬЗОВАНИЕ N-ГРАММ ДЛЯ СПЕЦИАЛИЗИРОВАННЫХ ТЕКСТОВ УВЕЛИЧИВАЕТ КАЧЕСТВО КАТЕГОРИЗАЦИИ.

В результате предварительной обработки будет составлен словарь, содержащий как отдельные слова, так и ключевые фразы. Элементы словаря далее будут называться терминами.

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

* 1. **Анализ основных методов (подходов) обработки текстов на естественном языке**

**Объяснение необходимости предобработки (2-ой вариант, первый по ссылке выше) https://cyberleninka.ru/article/n/povyshenie-tochnosti-bayesovskogo-klassifikatora-tekstovyh-dokumentov/viewer**

При том, что операциональная суть лингвистического разбора единооб-разна (выделяем компоненты и устанавливаем их свойства), в лингвистических экспертизах упоминается очень большое количество методов и список их носит практически открытый характер. Это многообразие связано с тем, на какие именно единицы разбирается сообщаемое. Именно тип единицы выступает как основной дифференциальный признак вида анализа (метода), он и фиксируется прежде всего в названии и описании метода.

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. **Семантическая декомпозиция как один из инструментов семантического анализа текстов**

использование нейронных сетей.

* 1. **Сравнительный анализ существующих алгоритмов семантической декомпозиции и выбор базовых алгоритмов для реализации**

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.

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. **ПОСТАНОВКА ЗАДАЧИ**

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 Теоретических основ разрабатываемых алгоритмов**
   1. **Алгоритм семантической декомпозиции на n-граммы**

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

* + 1. **Математические основы алгоритма**

дпдшмлм

* + 1. **Графическое описание алгоритма**

дпдшмлм

* + 1. **Сценарий применения алгоритма для решения задачи категоризации**

идпмдлид

* 1. **Алгоритм семантической декомпозиции на предложения**
     1. **Математические основы алгоритма**

дпдшмлм

* + 1. **Графическое описание алгоритма**

дпдшмлм

* + 1. **Сценарий применения алгоритма для решения задачи категоризации**

Идпмдлид

* 1. **Выводы**

адплпщжиди

1. **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ АЛГОРИТМОВ**
   1. **Алгоритм семантической декомпозиции на n-граммы**
      1. **Технические подробности программной реализации алгоритма**

лалалалал

* + 1. **Тестирование алгоритма**

лалалалал

* + 1. **Сценарий использования**

лалалалал

* + 1. **Применение алгоритма для решения задачи категоризации**

лалалалал

* 1. **Алгоритм семантической декомпозиции на предложения**
     1. **Технические подробности программной реализации алгоритма**

Лалалалал

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. **Тестирование алгоритма**

лалалалал

* + 1. **Сценарий использования**

Лалалала

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. **Применение алгоритма для решения задачи категоризации**

лалалалал

* 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. **Коммерциализация результатов исследований в рамках НИР**

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».

# **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.

Научная ценность данной работы заключается в улучшении качества результатов проекта автоматизации службы технической поддержки компании ООО ГПП.

# **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).