Страничка для титульного листа

1 Задание

Целью данной работы является продемонстрировать наработки сделанные по теме ВКР "Автоматическое извлечение ключевых слов/словосочетаний из документа на русском языке"по разделам: конструкторский, технологичеий и исследовательский во время прохождения преддипломной производтсвенной практики.

Для достижения поставленной цели необходимо выполнить следующией задачи:

1) заполнить конструкторский раздел;

2) заполнить технологический раздел;

3) провести серию эксперементов и занести результаты в иследовательский раздел.

2 Конструкторский раздел

В данном разделе приведено описание структуры алгоритма и отдельных его этапов. Представлена архитектура разрабатываемого программного продукта. Продемонстрирована диаграмма класов, использующихся в работе

2.1 Yake

Общая структура работы алгоритма по извлечению ключевых слов представлена на рисунках 2.1 - 2.2

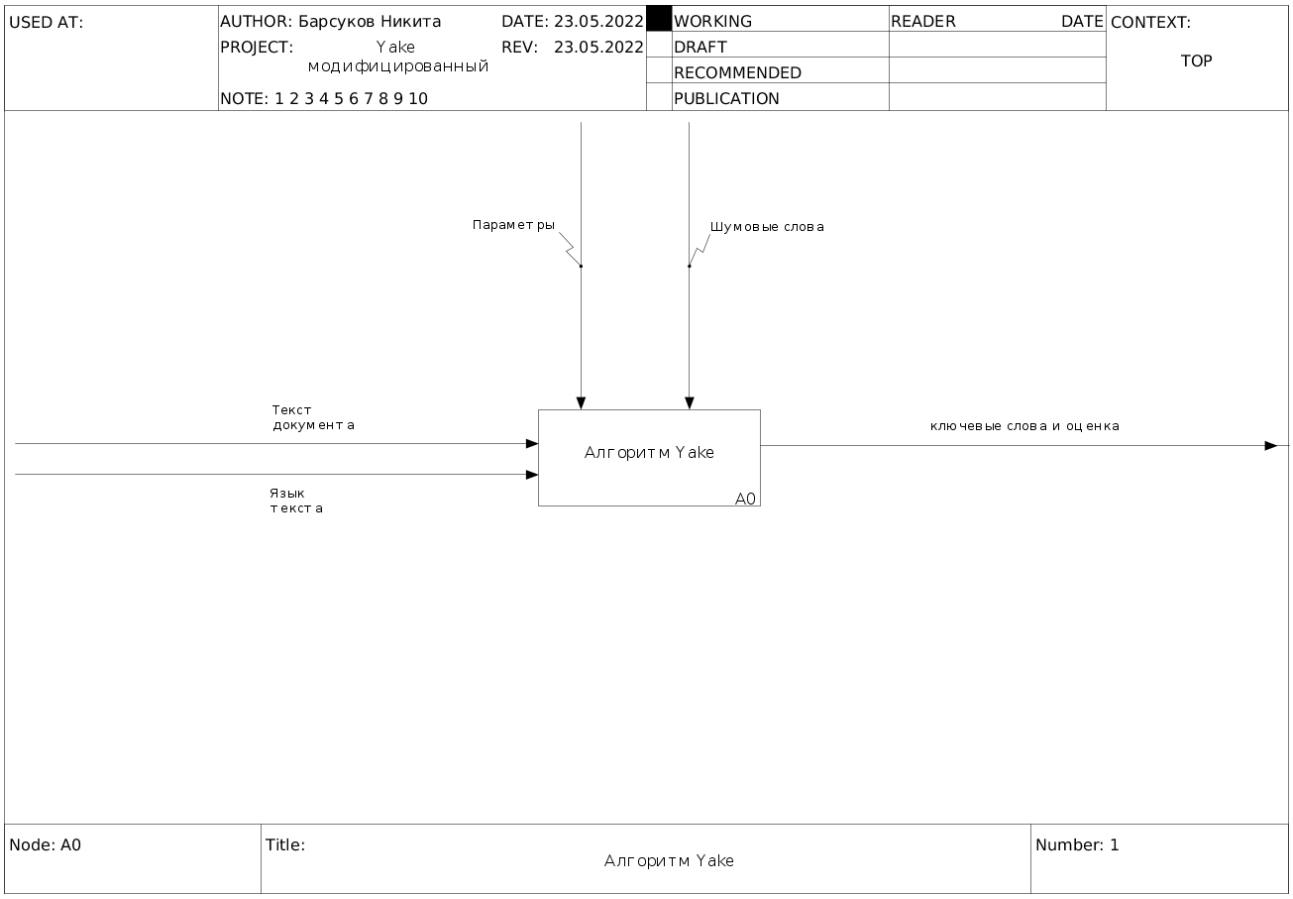

Рис. 2.1. IDEF0 диаграмма разрабатываемого метода

Входными параметрами данного алгоритма являются тест извлеченный из электронного докумета и язык текста. Резузьтатом исполнения данного метода является список, состоящий из кортежей, содержащих в себе термин и его оценку.

Данный метод можно разделить на несколько основных этапов, которые представлены на IDEF0-диаграмме, изображенной на рисунке 2.2

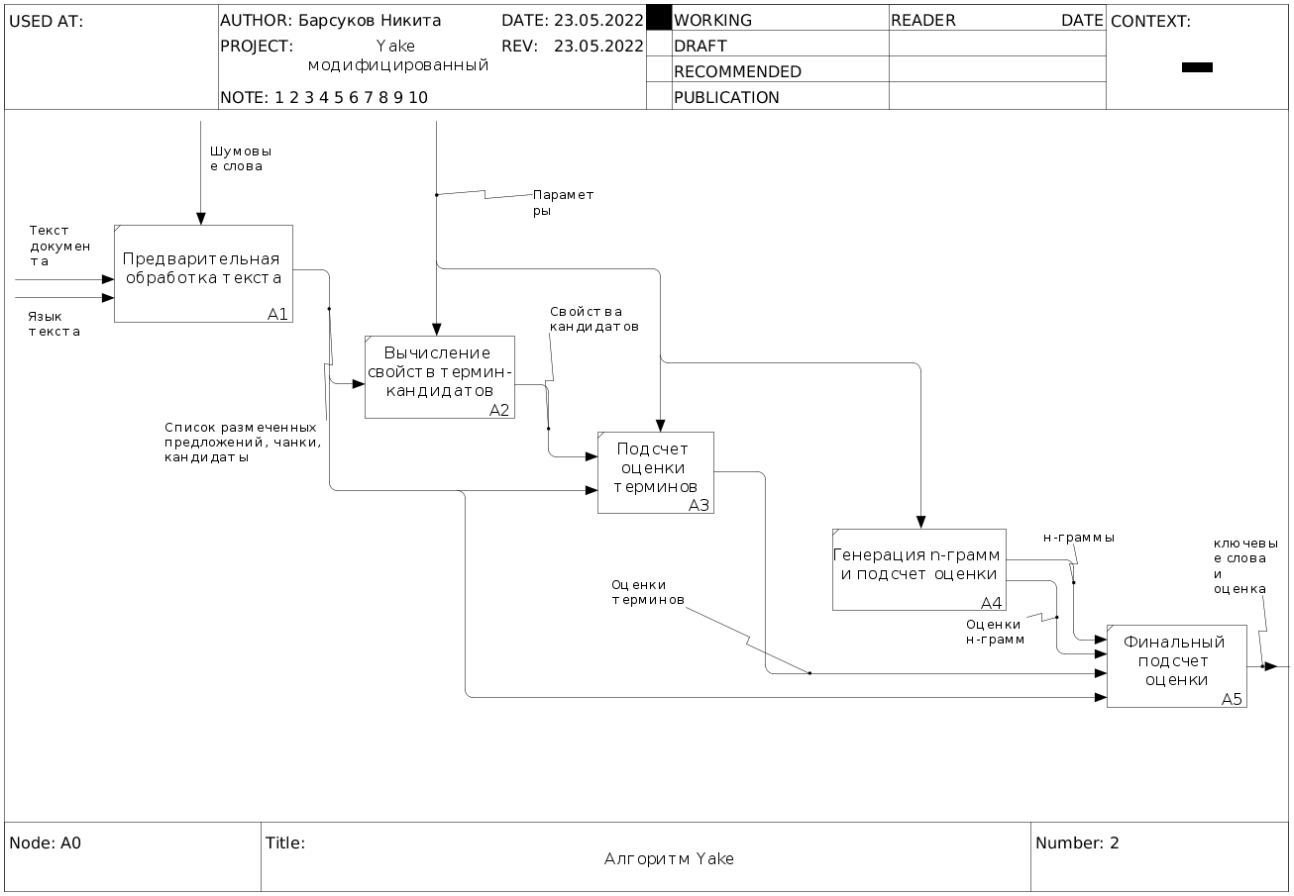

Рис. 2.2. IDEF0 диаграмма модуля извлечения ключевых слов из текста

1) предварительная обработка текста и выделение кандидатов (рисунок 2.2 блок A1)

2) вычисление свойств термин-кандидатов (рисунок 2.2 блок A2)

3) подсчет оценки кадидатов (рисунок 2.2 блок A3)

4) генерация н-грамм и подсчет оценки (рисунок 2.2 блок A4)

5) подсчет финальной оценки терминов (рисунок 2.2 блок A5)

2.2 Арихтектура ПО

Для реализации программного обеспечения была выбрана MVC архитектура, разбивающая программу на три отдельных компоненты:

1) представление - это отображение состояния внутренний системы;

2) модель - это компонента отвечающая за предоставление данных конкретным элементам систмы;

3) контроллер - это связующее звено между представлением и моделью, обрабатывает действия пользоватя, полученные от представления и отдает команды модели.

Через графический интерфейс у пользователю должна быть возможность взаимодействия с программный ПО. Под взаимодействием подразумевается запуск ПО и получения результата. Предоставляющийся функционал:

1) выбор из списка методов извелечения КС от одного до нескольких алгоритмов;

2) выбор одного или нескольких файлов через специальное окно;

3) установка параметров для методов в отдельном окне;

4) возможность ввести отдельно текстовую информацию.

На рисунке 2.3 схематично отображены основные классы разрабатываемого программного продукта

Класс MainWindow - это входная точка приложения. Он выполняет основную логику, связанную с обработкой пользовательский запросов к графическому интрефейсу. Он взаимодействует с классом MethodControler, отвечающего за подгрузку, подготовку методов к работе.

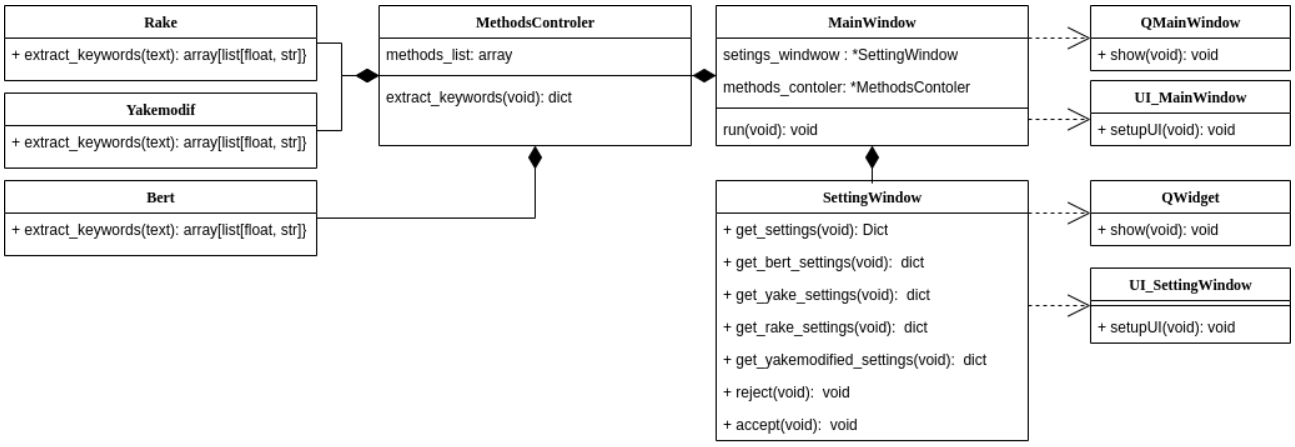

Рис. 2.3. Схематическое представление архитектуры программного обеспечения

3 Технологический

В данном разделе представлены средства, решения, а также технологии используемые при реализации данной работы. Приведено описание входных данных и продемонстрирован интерфейс программы.

3.1 Системные требования

Каждое программное обеспечение требует материальной базы в рамках которого оно будет функционировать. Описанием характеристик такой базы называются системные требования. Данные требования предоставляет пользователю информацию о аппаратном обеспечении, необходимого для использования ПО.

1) Операционная система: Ubuntu 18.04.6 TLS

2) Процессор: Intel(R) Core(TM) i5-8265U CPU @ 1.60GH 3900,00 MHz

3) Жесткий диск: 500 гб

4) Оперативная память: 16гб

3.2 Язык программирования

Продолжающиеся развитие компьютерных технологий и широкое их распространие спровоцировало спрос на развитие и создание новых знаковых систем для записи алгоритвом, которые известны на сегодняшний день как языки программирования (ЯП). Сформируем список из наиболее популярных и проведем обзор для установления их соответсвия поставленной задаче.

1) Python;

2) C++;

3) C#;

в данный список не попали языки относящиеся к веб разработке, по скольку разрабатываемое программное обеспечение представляет собой декстопное приложение.

3.2.1 Python

Python – это высокоуровневый язык программирования общего назначения, который ориентирован на повышение читаемости кода и производительности разработчика. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных в ходе разработки функций.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, императивное, аспектно-ориентированное и функциональное. Основными архитектурными чертами являются такие особенности как автоматическое управление памятью, динамическая типизация, механизм обработки исключений, полная интроспекция, гибкие высокоуровневые структуры данных и поддержка многопоточных вычислений. Код в Python организовывается в классы и функции, которые могут объединяться в модули, которые могут быть объединены в пакеты.

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, которая позволяет использовать его без ограничений в любых приложениях, включая проприетарные. Наиболее часто Python сравнивают с Ruby и Perl. Эти языки обладают примерно одинаковой скоростью выполнения программ и также являются интерпретируемыми. После проведенного сравнительного анализа современных языков программирования, для реализации программного обеспечения был выбран язык Python.

3.2.2 C++

C++ - это универсальный язык программирования. За исключением второстепенных деталей C++ является надмножеством языка программирования C. Помимо возможностей, которые дает C, C++ предоставляет эффективные и гибкие средства определения новых типов. Используя определения новых типов, точно отвечающих концепциям приложения, программист может разделять разрабатываемую программу

на легко поддающиеся контролю части. Такой метод построения программ часто называют абстракцией данных. Информация о типах содержится в некоторых объектах типов, определённых пользователем. Такие объекты просты и надёжны в использовании в тех ситуациях, когда их тип нельзя установить на стадии компиляции.

Программирование с применением таких объектов часто называют объектно-ориентированным. При правильном использовании этот метод даёт легче контролируемые программы, более короткие и проще понимаемые. C++ предлагает программисту полный набор операторов структурного программирования. Он также обладает очень большим набором операций. Многие операции C++ соответствуют машинным командам, и поэтому допускают прямую трансляцию в код ассемблера. Разнообразие операций позволяет выбирать их различные наборы для минимизации результирующего поля. C++ поддерживает указатели на переменные и функции. Указатель на объект программы соответствует машинному адресу данного объекта. Посредством разумного использования указателей можно создавать эффективные программы, которые выполняются быстро, так как указатели позволяют ссылаться на объекты тем же самым путём, как это делает машина.

3.3 C#

C# – это объектно-ориентированный язык программирования. Разработан в 1998-2001 годах в компании Microsoft в качестве языка разработки приложений для платформы Microsoft .NET Framework и впоследствии был стандартизирован как ISO/IEC 23270 и ECMA-334. C# относится к семье языков с C-подобным синтаксисом, среди них его синтаксис наиболее близок к C++ и Java. Язык поддерживает полиморфизм, перегрузку операторов (в том числе операторов неявного и явного приведения типа), имеет статическую типизацию, атрибуты, делегаты, свойства, события, обобщённые методы и типы, анонимные функции с поддержкой замыканий, итераторы, LINQ-запросы, комментарии в формате XML и исключения.

В результате сравнения языков программирования оптимальным

сочетание является Python и XMl. Python будет использовать для модулей логики и взаимодействия. Установлением связи между компонентами будет обеспечено с помощью применения слотов и сигналов из библиотеки PyQt5.

3.4 Формат файлов

Формат переносимых документов (PDF) представляет собой универсальный файловый формат, который позволяет сохранить шрифты, изображения и сам макет исходного документа независимо от того, на какой из множества платформ и в каком из множества приложений такой документ создавался. Формат Adobe PDF считается признанным общемировым стандартом в области тиражирования и обмена надежно защищенными электронными документами и бланками [24]

3.5 Библиотеки

Современное программное обеспечение состоит из множества компонентов и реализация всего самостоятельно увеличивало бы и без того продолжительный процесс разработки. Для решения данной проблемы используются библиотеки и фреймворки, которые предоставляют часть или полностью готовый функционал. Для реализации данной работы ипользовались следующие библиотеки:

1) numpy;

2) networkx;

3) segtok;

4) jellyfish;

5) poetry;

6) textract;

NumPy - это фундаментальный пакет для научных вычислений в Python. Это библиотека языка Python, которая предоставляет многомерный объект массива, различные производные объекты (такие как маскированные

массивы и матрицы) и ассортимент подпрограмм для быстрой работы на массивах, в том числе математический, логично, манипулирование формой, сортировка, выбор, дискретные преобразования Фурье, базовая линейная алгебра, основные статистические операции, случайное моделирование и многое другое.

jellyfish - представляет из себя набор функций для стемминга (процесса нахождения начальной формы слов), реалзиций методов нахождения редакционного расстояния таких как: расстояние Левенштейна, Дамерау-Левенштейна, Хамминга и Жаро.

Segtok - библиотека предоставляющая две моделя segmenter и tokenizer. Segmenter предоставляет функционал по разделению текста на Индо-Европейских язык на предложения. Tokenizer предоставляет инструмент для разибения предложений на слова и символы

Networkx - это библиотка для теории графов и средство моделирования сети, разработанное на языке Python, которое содержит встроенные графы и сложные алгоритмы сетевого анализа Используя networkx, позволяет хранить сети в стандартизированных и нестандартизированных форматах данных, генерировать различные случайные и классические, анализировать сетевые структуры, создавать модели сетей, разрабатывать новые сетевые алгоритмы и выполнять рендеринг сети. Networkx поддерживает создание простых неориентированных графов, ориентированных графов и мультиграфов; во многих стандартных алгоритмах теории графов узлами могут быть любые данные; поддерживается любое измерение граничных значений, функция Богатый и простой в использовании. В рамках работы используется для определение совместного появления терминов на этапе оценки свойств методов

Poetry - это инструмент для управления зависимостями и сборкой пакетов в Python. В Poetry представлен полный набор инструментов, которые могут понадобиться для детерминированного управления проектами на Python. В том числе, сборка пакетов, поддержка разных версий языка, тестирование и развертывание проектов.

3.6 Графический интерфейс

Для реализации графического интрефейса была выбрана библиотека PyQt5. PyQt5 - графическая библиотека для python, которая предоставляет возможность создавать графические интерфейсы для пользователя. Данная библиотека предоставляет объектно-ориентированные решения, которые включают в себя логическую иерархию между объектами, имеет понятную структуру наследования. Она является бесплатной, распространяется по лицензии GPL, LGPL. Для генерации xml схемы пользовательского интерфейса использовался QtDesigner с целью дальнейшего преобразования в класс интерфейса.

На рисунках 3.1 - 3.3 представлен интерфейс разрабатываемого ПО.

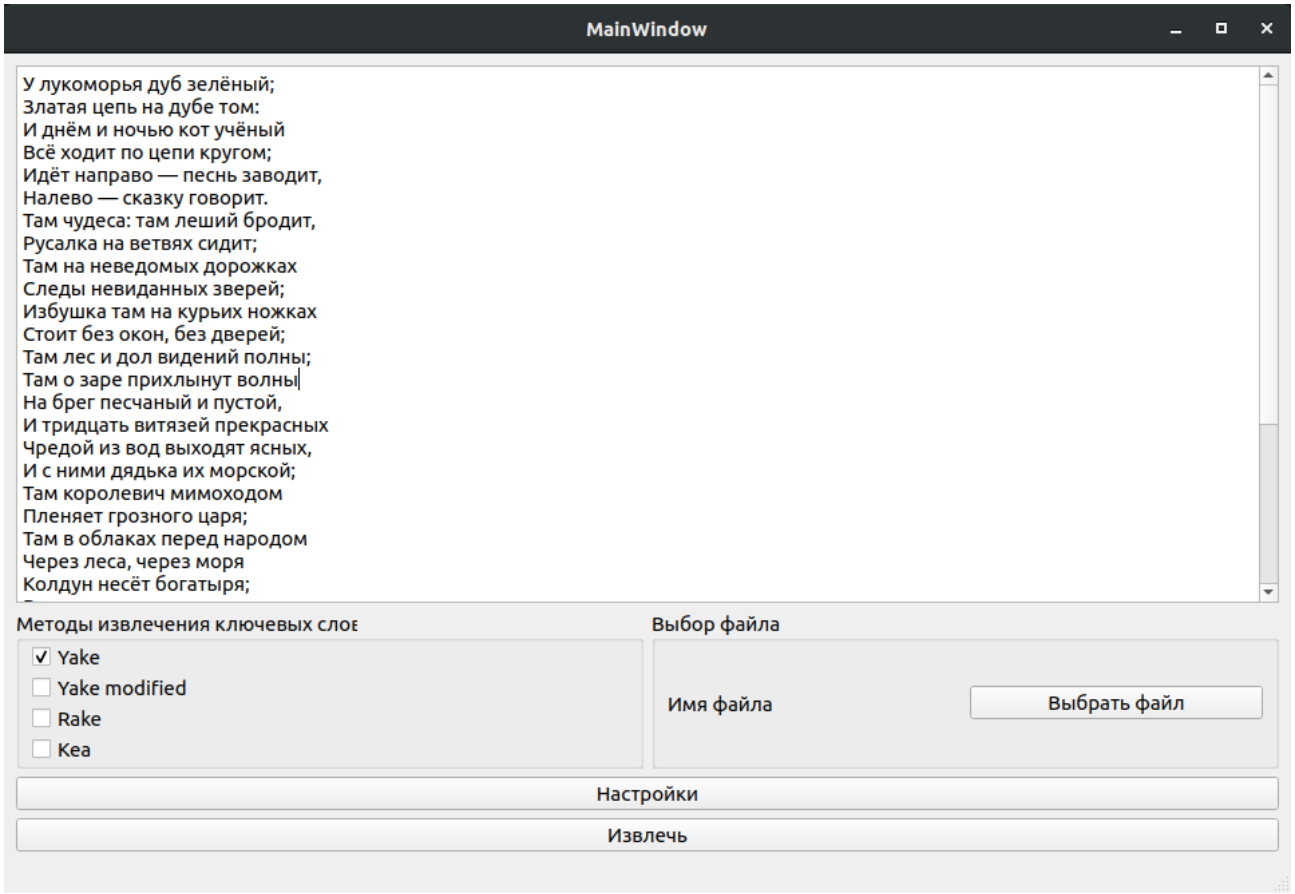

Рис. 3.1. Главное окно

Пользовательский интерфейс состоит из нескольких окон. На рисунке 3.1 представлено главное окно программы. В данном окне пользователь может указать фрагмент интересуемого текста, выбрать файл формата pdf для дальнейшего извлечение текстовой информации, выбрать алгоритмы,

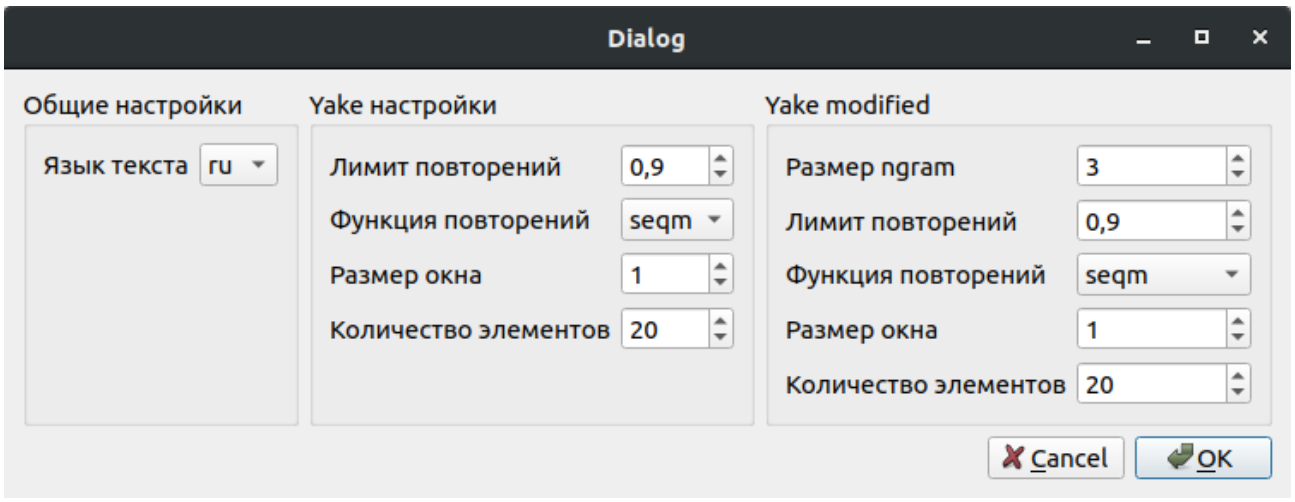

Рис. 3.2. Окно настроек методов

открыть окно параметров алгоритмов, изображенное на рисунке 3.2 и запустить процесс извлечения ключевых слов.

На вход ожидаются:

1) параметры методов;

2) документ формата pdf или тест

На выходе получем список кортежей состоящий из ключевых слов и оценок. Результат работы алгоритмы, отображен на рисунке 3.3. Для удобства проведения эксперементальной части окно результатов будет сделано к пред. дипломной защите.

3.7 Среда разработки

Интегрированная среда разработки или IDE (Integrated Development Environment) - специальный программный комплекс, предназначенный для полного цикла написания и тестирования программ на опредленном языке.

Интегрированная среда разработки облегчает работу, предоставляя программистам средства для разработки программного обеспечения, такие как редактор исходного кода, средства автоматизации сборки и отладчик. IDE облегчает визуальное представление файлов и делает его более понятным для пользователя.

Средой разработки для разработки ПО была выбрана IDE PyCharm от компании JetBrains, специлизирующейсы на производстве инструментов

keyword	score
Бабою Ягой Идёт	0.0024658574918241093
лукоморья дуб зелёный	0.0028145046290066335
ночью кот учёный	0.0035015896487106154
Златая цепь	0.004815232632714305
песнь заводит	0.008129438443169539
Идёт направо	0.0082670599665455 63
цепь на дубе	0.010290222515827 32
днём и ночью	0.010290222515827 32
ходит по цепи	0.010290222515827 32
видел дуб зелёный	0.010928958057816773
моря Колдун несёт	0.012134568952398 84
Следы невиданных зверей	0.013245708322672224
Пленяет грозного царя	0.013245708322672224
Колдун несёт богатыря	0.013245708322672224
дуб зелёный	0.013548004166796728
неведомых дорожках Следы	0.014709593335829656
курьих ножках Стоит	0.014709593335829656
витязей прекрасных Чредой	0.014709593335829656
королевич мимоходом Пленяет	0.014709593335829656
дорожках Следы невиданных	0.014709593335829657

Рис. 3.3. Результат работы алгоритма (демо)

для профессиональной разработки программного обеспечения. Данная среда была выбрана из за следующих удобст и преимуществ:

1) наличие полноценного отладчика как для кода так и для тестов.

2) встроенная подстветка синтаксиса

3) встроенный терминал;

4) интеграция с системой контроля версий (VCS) git;

5) поддержка множественных конфигураций запуска;

6) встроенный аналазитор классов;

3.8 Система контроля версий

Во время процесса разработки мною была использованна система контроля версий Git (https://git-scm.com). Система контроля версий с помощью репозиториев решает проблемы с переносом программного кода на другие устройства, его резервным копированием, а так же дает возможность разделения версий продукта во время разработки, что позволяет при внесении изменений или модифакциях всегда иметь рабочию версию проекта.

3.9 Вывод

В результате выполнения данного раздела были определены технические средства необходимые для разработки ПО. Описаны использующиеся библиотеки и приведена демонстрация пользовательского интерфейса.

4 Эксперементальный раздел

В данном разделе произведен ряд эксперементов с полученным, в ходе написания проекта, программным обеспечением. Для проведения серии эксперементов необходимо подготовить набор тестовых входных данных, представляющих собой тридцать электронных документов на русском языке, формата PDF. Все работы учавствующие в экперементе взята с сайта Для тестов отбираются тексты с указанными ключечевыми словами с целью сопоставления авториских КС с полученными из метода.

Для обеспечения качества проводимых эксперементов необходима указать критерии по которыму будет проиводиться отбор документов:

1) документ должен содержать в себе текст, а не отсканированные изображения страниц ранее опубликованных работ, по скольку это приводит к невозможности прочтения документа;

2) информация содержащаяся в документе должна быть целой, то есть принадлежать одной работе.

4.1 Исследование зависимости результата от параметра n

Целью данного исследования является изучение зависимости результата извлечения ключевых слов, от параметра n, отвечающего за размер используюемых n-грамм. На рисунке 4.1 представлены результаты работы алгоритма при n варьирующемся от 1 до 3.

Как и предпологалось в зависимости от параметра n будут появляться многокомпонентные термины, размер которых зависит от данного параметра. При n = 1, однокомпонентные КС, при n = 2 двухкомпонентные, при n = 3 трехкомпонентные. Примером расширения ключевого слова является "информационнотелекоммуникационной"который преобразуется в "информационнотелекоммуникационной сети"и "мобильной информационнотелекоммуникационной"(рисунок 4.1), а затем в "мобильной информационнотелекоммуникационной сети"

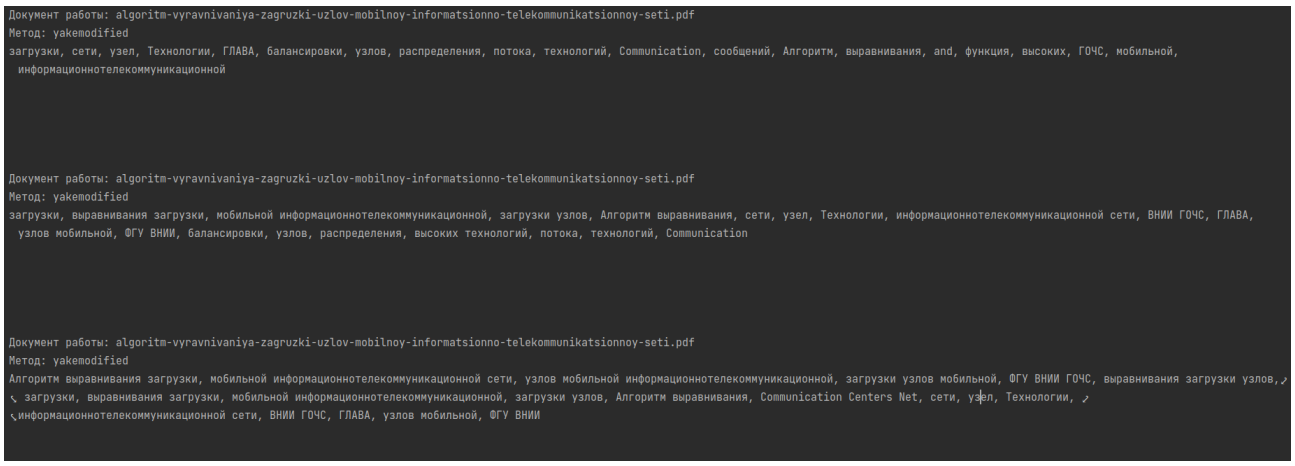

Рис. 4.1. Результат работы алгоритма от параметра n

4.2 Исследование точности метода

В рамках данного эксперемента оценивается результат работы модифицированного метода Yake на ранее подготовленных тестовых данных, путем оценки процентного пересечения ключевых слов отмеченными авторами текста с КС полученными в итоге отработки метода.

Для алгоритма Yake были выставлены параметры отображенные на рисунке 4.2

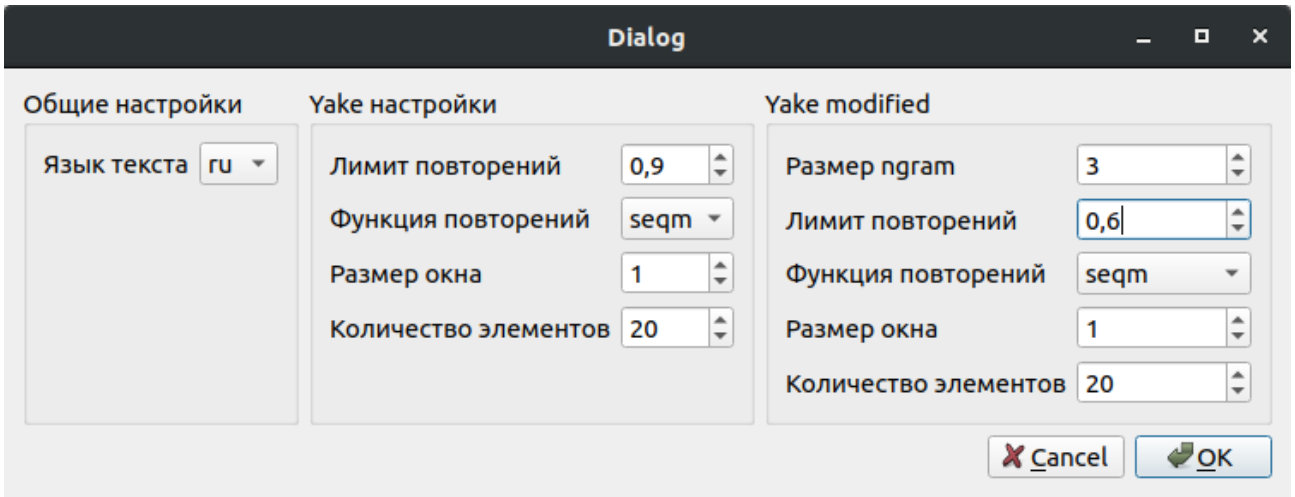

Рис. 4.2. Параметры модифицированного метода Yake

На рисунке 4.3 представлен результат сравнения ключевых слов предоставленных авторами документов с КС полученные путем извлечения. В ходе данного эксперемента было установлено что процент пересечения оригинальных слов и полученных путем извлечения составляет 33%.

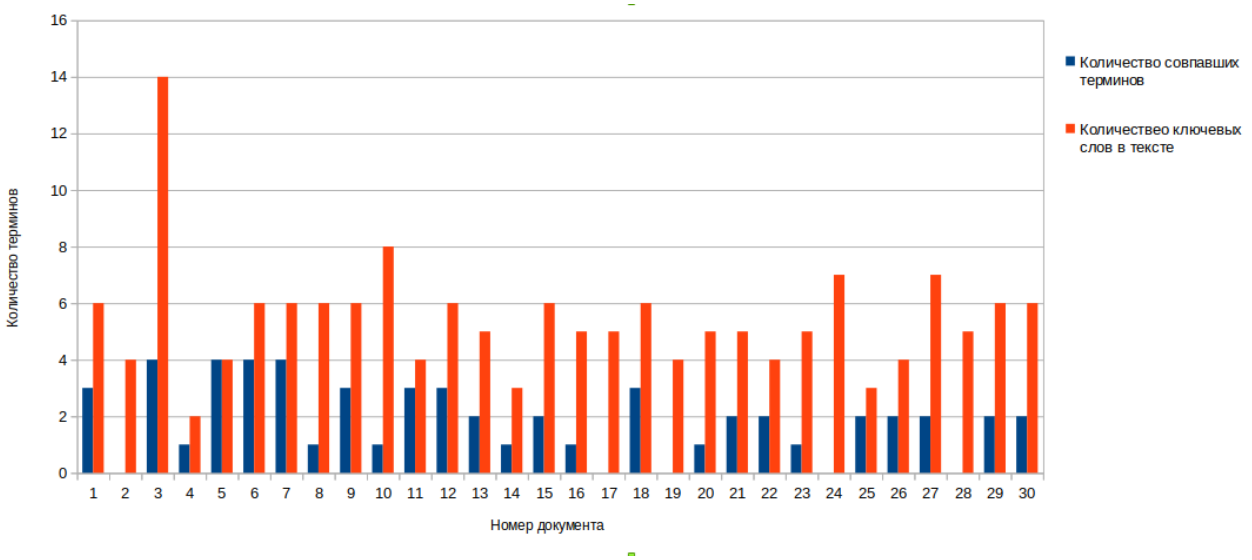

Рис. 4.3. Результат пересения ключевых слов

4.3 Вывод

В результате проведения исследовательской работы над разработанным решением было установлено, что все требования, поставленные к алгоритму, соблюдены. Алгоритм способен на извлечение многокомпонентных ключевых слов из документов на русском языке.

5 Заключение

По итогу выполненния работы была спроектирована архитектура, разработано и протестировано программное обеспечение, осуществляющие извлечение ключевых слов из документов на русском языке. Помимо выше сказаного были проведены иследования, продемострировашие пригодность метода по работе с документами на русском языке.

В процессе выполнения работы:

1) проанализирована предметная область и произведена классификация существующих методов КС;

2) разработана архитектура программного обеспечения;

3) проведена модификация метода и разработы модули;

4) произведено исследование получившегося ПО.

Из достоинства полученного метода можно выделить следующее:

1) не требует обучение;

2) не требует наличие корпуса текстов;

3) возможность извлечения n-компонентных ключевых слов;

4) не использует тезаурусы.

Из возможных путей развития стоит отметить:

1) добавление в предварительную обработку процесса стемминга;

2) добавить авто авпределение языка;

3) обучить алгоритм определять синонимические термины.

Список использованных источников

1. YAKE! Keyword extraction from single documents using multiple local features // URL: *https* : *//www.sciencedirect.com/science/article/abs/pii/S*0020025519308588 (Дата обращения 12.01.2022)

2. Textual Analysis: A Beginner's Guid // URL: *http* : *//www*1.*cs.columbia.edu/ sbenus/Teaching/APTD/McKee*$_C$*h*1.*pdf* (Дата обращения 26.01.2022)

3. Automatic keyphrases extraction based on NLP and statistical methods // URL: *https* : *//www.researchgate.net/publication/*220827238$_A$*utomatic*$_K$*eyphrase*$_E$*xtractio* (Дата обращения 08.02.2022)

4. Keyword extraction from a single document using centrality measures // URL: (Дата обращения 08.02.2022)

5. Michael W. Berry Text Mining Application and Theory

6. А.Н. Соколов Внутренняя речь и мышление // URL: *https* : *//search.rsl.ru/ru/record/*01008431174 (Дата обращения 08.02.2022)

7. Современные методы автоматизированного извлечения ключевых слов из текста

8. Automatic keyword prediction using Google similarity distance // URL: *https* : *//www.sciencedirect.com/science/article/pii/S*0957417409006745

9. Методы и модели автоматического извлечения ключевых слов // URL: *https* : *//cyberleninka.ru/article/n/metody − i − modeli − avtomaticheskogo − izvlecheniya − klyuchevyh − slov*

10. (Для понимания TF-IDF) Understanding Inverse Document Frequency: On Theoretical Arguments for IDF URL: *https* : *//www.researchgate.net/publication/*238123710$_U$*nderstanding*$_I$*nverse*$_D$*ocume*

11. A statistical interpretation of term specificity and its application in retrieval // URL: $http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.115.8343\&rep=rep1\&type=pdf$

12. TF-IDF in a nutshell // URL: $https://towardsdatascience.com/tf-idf-in-a-nutshell-b0ff082fbbc$

13. // URL: $https://courses.ischool.berkeley.edu/i256/f06/papers/luhn58.pdf$

14. YAKE // URL: $https://www.sciencedirect.com/science/article/pii/S0020025$

15. Automatic Keyword Extraction from individual Document // URL: $https://www.researchgate.net/publication/227988510_Automatic_Keyword_Extraction_$

16. История интернета // $https://sites.google.com/site/globalnyekomputernyese$ $i-kak-poavilsa-internet$ (Дата обращения 10.03.2022)

17. Информационная революция // $https://bigenc.ru/technology_and_technique/text/2015889$ ()

18. PositionRank: An Unsupervised Approach to Keyphrase Extraction from Scholarly Documents // $https://aclanthology.org/P17-1102.pdf$

19. The Automatic Creation of Literature Abstracts // $https://ieeexplore.ieee.org/document/5392672$

20. Universal Mobile Information Retrieval $https://link.springer.com/chapter/10.1007/978-3-642-02710-9_38$

21. Исследование метода выделения однословных терминов в тематических текстах // $https://cyberleninka.ru/article/n/issledovanie-metoda-vydeleniya-odnoslovnyh-terminov-v-tematicheskih-tekstah/viewer$

22. N-грамммы в лингвистике $http://lab314.brsu.by/kmp-lite/kmp2/Translation/MT/MT-Corpus/n-grammy-v-lingvistike.pdf$

23. МНОГОКОМПОНЕНТНЫЕ ТЕРМИНЫ СФЕРЫ ТЕПЛОЭНЕРГЕТИКИ // $https://elar.urfu.ru/bitstream/10995/60427/1/975-8295-0572-1_2018_150.pdf$

24. Portable Document Format (PDF) *https* : *//helpx.adobe.com/ru/incopy/using/pdf.html*