МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВО «КубГУ»)**

**Факультет компьютерных технологий и прикладной математики**

**Кафедра информационных технологий**

Допустить к защите

Заведующий кафедрой

канд. техн. наук, доц.

\_\_\_\_\_\_\_\_\_ В.В. Подколзин

(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

**(БАКАЛАВРСКАЯ РАБОТА)**

**СЕРВИС** **ОПРЕДЕЛЕНИЯ СООТВЕТСТВИЯ ТЕХНИЧЕСКОЙ ДОКУМЕНТАЦИИ ГОСТу**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Д. А. Арабова

(подпись)

Направление подготовки 01.03.02 Прикладная математика и информатика

(код, наименование)

Направленность (профиль) Программирование и информационные технологии

Научный руководитель

канд. пед. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Н. Ю. Добровольская

(подпись)

Нормоконтролер

канд. пед. наук, доц. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. В. Харченко

(подпись)

Краснодар

2024

**СОДЕРЖАНИЕ**

Введение 3

1. Анализ текста на естественном языке 5

1.1. Выделение базовых задач анализа текста 5

1.2. Технологии обработки естественного языка 9

1.3. Реферативный обзор программ интеллектуального анализа

текста 12

1. Задача анализа технической документации 15

2.1. Виды технической документации 15

2.2. О существующих программах проверки соответствия ГОСТу 18

1. Проектирование сервиса определения соответствия технической документации по ГОСТу 20

4. Особенности реализации сервиса и примеры работы 29

Заключение 39

Список использованных источников 40

Приложение А Шаблоны технической документации «Описание применения», «Руководство оператора», «Текст программы» 42

**ВВЕДЕНИЕ**

В современном мире интеллектуальный анализ текста занимает все большие области человеческой деятельности, начиная от автоматического реферирования научных статей и заканчивая эмоциональной оценкой сообщений в социальных сетях. Многочисленные алгоритмы анализа текстов основаны на машинном обучении и призваны минимизировать участие человека в решении подобных задач.

Вообще, естественный язык – это язык, применяемый для коммуникации людей в обществе, который создан не специально. Однако, алгоритмы анализа текстов могут применяться и к анализу технической документации, которая составляется на естественном языке, но имеет строго определенную структуру.

Техническая документация представляет собой совокупность программных и технических документов, используемых при разработке, строительстве и эксплуатации зданий, сооружений, программного и аппаратного обеспечения.

Все программные продукты содержат техническую документацию, которая имеет строгую структуру и содержание, определяемые ГОСТами. Для людей, составляющих эту документацию, одной из важнейших задач является соблюдение требований. Для того чтобы минимизировать человеческий труд, при проверке корректности структуры документов, необходимо использовать все возможности компьютерной лингвистики, позволяющей автоматизировать решение задачи.

Первая глава выпускной квалификационной работы посвящена выделению базовых задач анализа текста, изучению технологий обработки естественного языка и проведению реферативного обзора программ интеллектуального анализа текста.

Во второй главе приводятся виды технической документации, их особенности и рассматриваются существующие программы проверки соответствия ГОСТу.

Третья глава содержит построение различных диаграмм, определение задач и требований к ним.

В четверной главе описаны выделенные при изучении ГОСТов пункты для проверки файлов технической документации, приведены примеры разработанных функций и примеры работы сервиса.

**1 Анализ текста на естественном языке**

* 1. **Выделение базовых задач анализа текста**

Имеется в сове время разработанная методика анализа текста, которая представляет собой один из праксиметрических методов, которые в свою очередь являются составляющей частью общенаучных методик, представляющих из себя измерение, эксперимент и наблюдение.

Особенностями текста, являющегося объектом исследования, обусловлен сам подход к его анализу и выбранный способ среди других методов праксиметриии. Суть текста многогранна, в связи с этим важно обратить пристальное внимание на его культурологические, психологические, лингвистические и другие стороны, для того чтобы реализовать обработку текста, а в частности текста с непростой системой коммуникативных единиц, в котором структура предложений и слов, располагающей целостностью, зависимостью между собой и имеющих некоторую степень завершенности.

Определение и синтез речи по тексту.

Существует множество сильных сторон сферы анализа естественного языка, главной из которых считается синтез и идентификация произносимой речи. Для решения поставленного вопроса, определения цифр, произнесенных человеком, в 1952 году, создано первое в своем роде устройство с помощью, которого цель достигнута. В современной действительности присутствует не которое количество фундаментальных методик определения устной речи.

Фундаментом одного из способов считается определение отдельных команд, которые являются элементами предварительно зафиксированного словаря, что в свою очередь влияет на показатель эффективности и степень точности идентификации. Одной из сфер использования такого метода является голосовое управление на сайтах. Система голосового самообслуживания включает в себя методику распознавания выражений, которые удовлетворяют некоторым грамматическим нормам [1].

Идентификация важных слов в тексте естественной речи, где аппаратно определяются те ее элементы, в которых располагаются определенные словосочетания или слова, что описывает третий метод, который эффективно внедрен в мониторинг речи и в системы поиска.

Совокупность методов идентифицирования слитной речи на объемном словаре, определяемая как STT- speech to text, стремится к переводу всей воспроизведённой речи в текстовый вид. В наше время конченое исполнения вопроса, который поставлен не достигнуто, но варианты активно внедряются, так как точность идентификации естественной речи определенно высока.

Обеспечение внесения текста на электронно-цифровые носители.

Создание сервисов, приложений, модулей, которые позволяют с высокой точностью орфографические, синтаксические, речевые ошибки в тексте, производят автоматический перенос слова – это и есть смысл и суть комплекса программ ввода текста на электронно-цифровые носители. При помощи этих задумок появляется возможность нахождения морфологических, лексических, синтаксических ошибок, с последующим внесением правок, а если еще точнее исключение ошибок, то есть будет происходит распознание данных в списке позволенных структур, либо будет искать ближайшее совпадение. Задачей системы является не только распознание написанного и напечатанного текста, но и автозавершение.

Машинный перевод.

Методика перевода, разработанная более 50 лет назад, брал за базу обычный способ словесного перевода. Однако незначительная эффективность и точность машинного перевода, вероятно находит объяснение в неполноте лингвистической модели, которая охватывает полностью весь язык, составляющими которого фактически являются семантика и прагматика. Имея в текущий момент множество всевозможных комплексов компьютерных программ перевода, разного уровня эффективности спустя многие годы разработок эффективного решения машинного перевода нет.

Поиск информации.

Определенные ключевые слова способствуют индексации текста, что в свою очередь входит в состав поискового индекса документа. Суть и предмет документа хорошо передают словосочетания, которые принимают за ключевые слова. В течение индексации документов выявлены трудности, которые повлекли за собой заключение, смысл, которого в том, что для идентификации приоритетных словосочетаний нужно задействовать всяческие нормы статистики и лингвистики.

Составлена реферата и анонсирования текста.

Первым шагом является разделение текста на высказывания или их элементы, вторым шагом распознаются значимые высказывания эти шаги являются этапами нахождения решения поставленного вопроса.

Выполнение сжатия текста и написание изложения меньшего по размеру по типу реферата называется реферированием текста. Кроме этого, представляется возможным создать для нескольких документов единый реферат, которые объединят одна тематика изложения. Аппаратное составление реферата заключается в идентификации приоритетных предложений в тексте, а после вычисление процента их важности. Формирование реферата и создание аннотации к тексту документа имеют общие черты. Главные темы, которые поднимаются в тексте, входят в состав аннотации в легком ее воспроизведении, что требует внедрения технологий индексирования.

Текстовая классификация.

Для создания больших пакетов документов возникают задачи текстовой классификации и кластеризации, итоговой целью которых является формирование близких по смыслу перечней документов. Опираясь на частные характеристики, производится отнесение отдельного документа к некой группе, что представляет собой классификацию, а смыл кластеризации в свою очередь заключается в том, чтобы разбить список документов на кластеры, которые включают в себя подгруппы документов близких по смыслу. Текст, сортируемый на конкретные смысловые рубрики, близок к текстовой классификации в этом и заключается задача рубрицирования. Значимость классификации увеличивается за счёт внедрения ее при распознавании спама, а применение этих методик при сортировке SMS сообщений является не новым направлением.

Знания и факты, полученные из текста.

Знания, добытые из текста, представляют собой одну из наиболее главных технологий анализа данных в производственном и экономическом разделах, а именно нахождение названных предметов и всевозможных сущностей и событий, имеющих отношение и связь с ними. Синтаксический текстовый анализ чаще всего входит в этот процесс и делает возможным для внедрения его в обработке различного рода новостей и информации от всевозможных агентств.

Исследование и анализирование правовых текстов.

Все существующие нормативные тексты – это иски, законы, постановления, планы работы, соглашения, договоры, уставы и другие подвергаются анализу с целью определения всяческих логических ошибок и пропусков. К примеру, осуществление анализа отдельно взятого закона выявило: требуемые мероприятия донесены не в полной мере, включая действующие лица и нормы отчётности, из чего следует, что проверяемый закон, не покажет необходимого уровня результативности.

Question Answering (системы вопросов и ответов) .

Нахождение необходимых текстов, в которых имеются ответы на сформулированные задачи и извлечение сведений из них, а в частности конкретизация типа запроса способствует получению ответа на поставленный вопрос. Составление реферата, распределение по классам и кластерам информации, задачи на которые найдено решение, распознавание и индексирование предметов и иные методики определяют задачи в технологическом поле Text Minig.

Диалог с компьютерными системами на естественном языке.

Необходимая цель достигается с помощью баз данных отдельного назначения. Внутри них у языка запросов существуют определенные лексические и грамматические ограничения, благодаря чему представляется доступным эксплуатация примитивных моделей языка. Переведя все запросы с естественного языка на формальный, проводится поиск нужной информации и формирование соответствующего ответа.[2].

* 1. **Технологии обработки естественного языка**

Общепринятые методы машинного анализа текста.

1. Способ автоматического анализа текста содержит в себе токенизацию, которая подразумевает под собой деление текста на слова и токены, которые изначально определены. Благодаря использованию этого процесса в качестве основы анализирования текста появляется возможность преобразить текст в упорядоченные формы, которые помогут при дальнейшего анализе.

2. Лемматизаией и Стеммингом именуется методика нормализации слов, приводящая слова к их корневой форме. Стемминг убирает аффиксы слов, а лемматизация приводит слова к инфинитивам их формы и проводит морфологический анализ.

3. Если текст можно представить численным вектором, любая составная часть которого сообщат четко обозначенную характеристику текста, то такой текст зовётся векторным представлением. "Мешок слов" и Методика статистического анализа являются распространёнными методами этого способа.

4. Кластеризация — это процесс объединения текстов в группы по их свойствам [3].

Методы машинного обучения

1. Наивный Байесовский классификатор – несложная и мобильная технология классификации текстов, база которой основана на пользовании теоремы Байеса. Этот способ довольно быстр и прост и благодаря ему можно распознать язык текста, разместить фильтр спама и разбить новости по сферам.

2. В методе опорных векторов или задачах классификации и регрессии осуществляются с помощью методов машинного обучения. Support Vector Machines во время выполнения анализа текстов при обработке большого количества признаков демонстрирует значительную точность.

3. Среди множества разных технологий регрессии и классификации есть методики основанные на образовании иерархии решений, такими являются методы решающие деревья и случайный лес. Комплекс деревьев обеспечивает высокую точность в способе случайный лес, а для эффективной трактовки и поиска применения для выявления правил из текстов используются решающие деревья [4].

Методы глубокого обучения

1. Структура нейронных сетей, называемая свёрточной нейронной сетью, разрабатывалась для анализа изображений и для эксплуатации в анализе текста. При анализе тональности, распределении текстов на классы для большой результативности в выводе итога используется Convolution neural network.

2. Созданы необычные рекуррентные нейронные сети, анализирующие последовательные данные, в частных случаях текста, который является структурой нейронных сетей. Автоматический перевод, анализ речи, создание текста проводятся благодаря таким нейронным сетям, так как у них есть способность определить взаимосвязь слов и зависимости предложений в тексте.

3. Передовые структуры нейронных сетей, полученные, благодаря технике внимания, называемые трансформерами и моделями на основе Bidirectional Encoder Representations from Transformers. Анализ тональности, поиск информации, а также системы «Вопрос - Ответ» являются целями машинного анализа текста и проводятся с большой результативностью, обеспечиваемой моделями, использующими технологии BERT, в свою очередь трансформеры достигают высоких результатов при обработке значительных объёмов текста [5].

Предобученные модели и трансферное обучение.

Методом машинного обучения, основанным на использовании информации и совокупности знаний и навыков, сложенных в итоге выполнения одной задачи, в течение выполнения иных задач именуется трансферным обучением.

Существуют предобученные модели, эффективно вводятся для достижения всевозможных целей без нужды обучения с нуля, что дает шанс вследствие трансферного обучения в случае при машинном текстовом анализе такими видами, являются GPT, ROBERTA и другие.

Естественный язык и его анализирование (NLP).

Область науки, в которой создаются комплексы методов и способы текстовой обработки, восприятие и написание текста на естественном языке именуется анализа естественного языка. Благодаря интеллектуальным системам, которые получены с применением Natural Language Processing, включающие в себя знания и технологии машинного обучения, лингвистики и других сфер происходит общение с людьми на естественном языке [6].

Различные технологии и методики машинного текстового анализа относятся к элементам NLP таким как:

* parsing – разбор предложений
* machine translation – машинный перевод
* semantic parsing – семантический анализ
* Part-of-Speech tagging и POS-tagging – определение частей речи
* sentiment analysis – анализ тональности
* text summarization – суммаризация текста
* Named Entity Recognition и NER – извлечение сущностей и др.

Текст, обрабатываемый системами, первоклассно и сложно созданными, нужные, реализующие разбор, понимание и точный текстовый анализ, являются полезны для владения методиками и способами NLP молодым разработчикам [7].

* 1. **Реферативный обзор программ интеллектуального анализа текста**

Рассмотрим примеры сервисов, осуществляющих интеллектуальный анализ текста.

Для первого примера возьмем веб-проект Voyant Tools (рисунок 1).

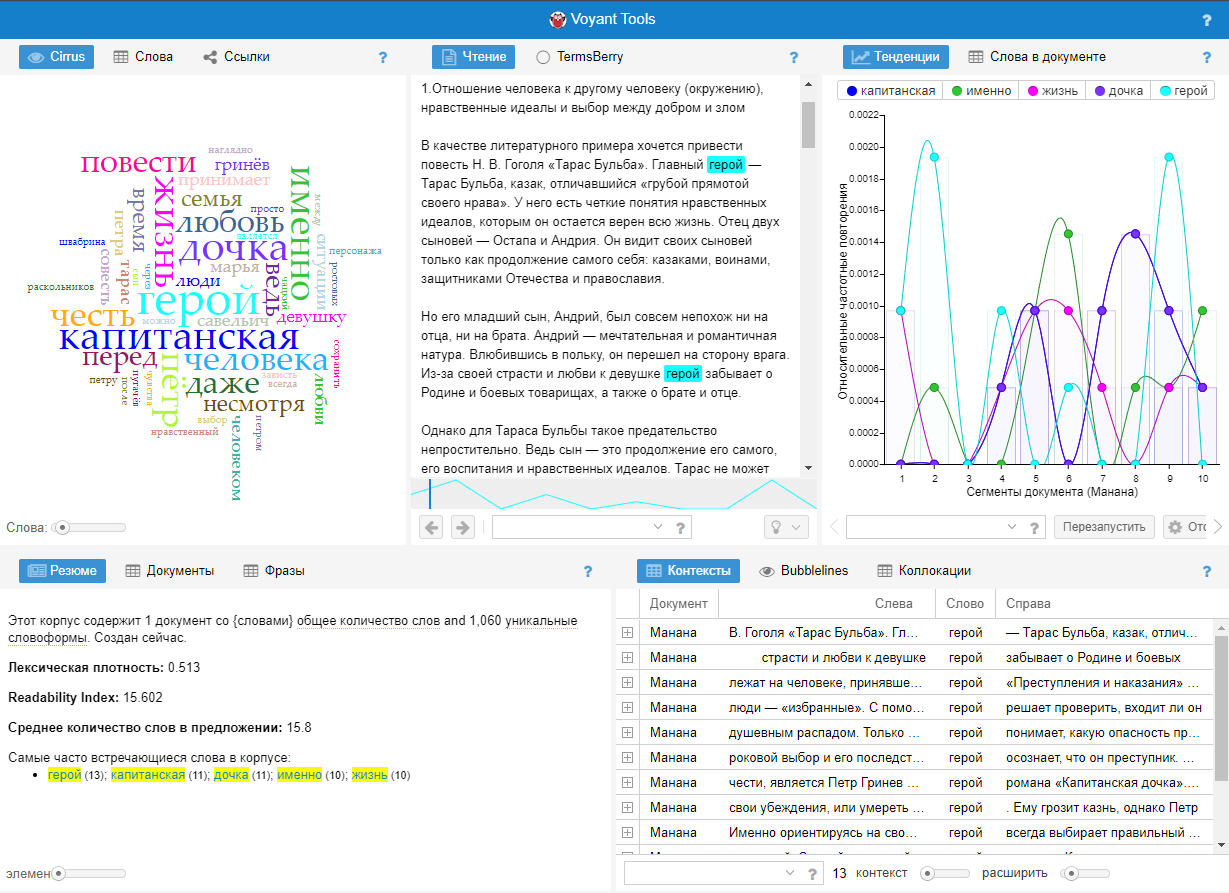


Рисунок 1 – Пример работы веб-проекта

На рисунке 1 показана работа веб-проекта с текстовым файлом, в котором находился текст сочинения. Этот сервис может работать не только с файлами формата docx, но и xml, pdf, rtf, doc, можно непосредственно в поле ввода вставить сам текст, также напрямую указать URL-адрес. Веб-проект обработал файл, посчитав количество слов, их частоту, лексическую плотность, среднее количество слов в предложении, график распределения частотности и анализ ключевых слов.

После обработки текста, рассматриваемый сервис предоставляет возможность сохранения результатов в различном формате: tab-separated, JSON и в виде визуализаций.

Вторым примером рассмотрим программное обеспечение Lexalytics (рисунок 2).

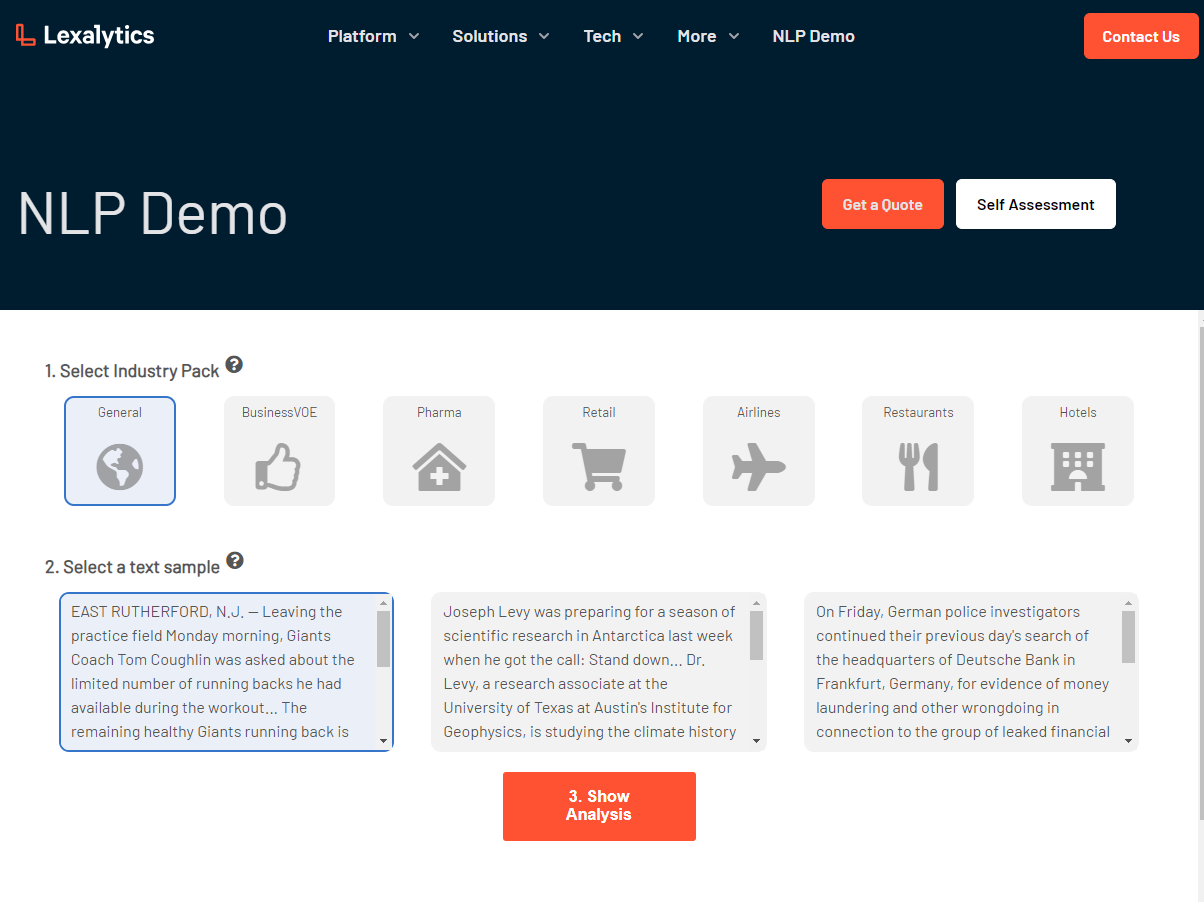
****

Рисунок 2 – Lexalytics

На рисунке 2 демонстрируется одна из страниц сайта программного обеспечения, обрабатывающего текст, применяя технологии обработки естественного языка (NLP). На сайте можно воспользоваться демоверсией ПО. Сначала необходимо выбрать отраслевой пакет (Industry Pack), затем выбрать пример текста (Text sample). После обработки текста можно ознакомиться с резюме по тексту, с областью классификацией текста, выделенной с использованием четырех методов, с темами текста, которые подбираются с помощью поиска фраз, соответствующих шаблонам POS и определенными в тексте сущностями: люди, места, даты, компании, продукты, вакансии, названия и хэштеги.

Основными функциями программного обеспечения Lexalytics являются анализ тональности, категоризация, извлечение сущностей и интенациональное обнаружение.

**2 Задача анализа технической документации**

**2.1 Виды технической документации**

Список документов, точно установленных для использования в процессе проектирования, разработки и использования технических объектов, в перечень в том числе входят построения, сооружения, аппаратное и программное обеспечение и другие именуются технической документацией.

Технологическая документация и ее тип:

* документы, предоставляющие информацию, необходимую для организации ремонта и производства изделия,
* технологическая документация,
* документы, определяющие технологический цикл изделия,
* ремонтная документация,
* конструкторская документация,
* эксплуатационная документация.

Разделяют два типа документации программного обеспечения:

* программная документация,
* эксплуатационная документация.

Техническая документация показывает и дает возможность наблюдать за правильностью течения процесса, вовремя найти отклонения или нарушения, и не допустить выпуск некачественной продукции или оказания услуг.

Кроме этого, техническая документация нужна при оформлении сертификатов соответствия, договоров и для исполнения надзорными органами инспекционных проверок в компании.

Совокупность документов, содержащая информацию, нужную для разработки, изготовления, эксплуатации и поддержки программы является программной документацией.

Типы документов, являющиеся программной документацией:

* пояснительная записка,
* описание программы,
* эксплуатационные документы,
* программа и методика испытаний,
* текст программы,
* техническое задание [8].

Для корректного и высокоэффективного применения и работы программы нужен список документов, относящихся к эксплуатационной документации.

Эксплуатационная документация и виды ее файлов:

* руководство по техническому обслуживанию,
* руководство системного программиста,
* руководство программиста,
* описание применения,
* описание языка,
* руководство оператора,
* ведомость эксплуатационных документов,
* формуляр.

Подробнее изучим структуру конкретных видов технической документации, а в частности описание применения, программный код, описание программы, руководство системного программиста.

Любой из перечисленных документов обязан соответствовать стандартам ГОСТ и содержать обязательные части. Программная документация содержит условные разделы:

* титульной,

В титульную часть входит: титульный лист и лист утверждения.

Требования ГОСТ 19. 104–78 полностью описывают оформление титульного листа и листа утверждения [8].

* информационной,

В информационную часть входит аннотация и содержание. Информационная часть безусловно должна присутствовать в разных видах программной документации, это определенно конкретными нормами на них единой системы программной документации. Краткое содержание основной части документа и его цели составляют аннотацию.

Состав оглавлений, включенный в содержание, довольно детально описывает структурный элемент в главной части документа, когда определяется обозначение (раздел, подраздел и другие), название, адрес на носителе информации (номера страниц, файлов и прочее) для этого элемента

* основной,

Содержание и структура главной части строго организована в технической документации, что определяется единой системой программной документации для определенных его типов.

* регистрации изменений [8].

В ГОСТе 19.603–78 указано, что любая поправка в этой части программного документа обязательно должна быть отмечена.

Этот стандарт фиксирует общие требования к виду программной документации для систем и комплексов вне зависимости от предназначения и сфер использования и для вычислительных машин, учитывая требования, соответствующие регламенту единой системы программной документации, требования стандартов, которые берут во внимание любые способы написания документов на всевозможных носителях.

* 1. **О существующих программах проверки соответствия ГОСТу**

Рассмотрим примеры сервисов, осуществляющих проверку соответствия ГОСТу.

Для первого примера возьмем онлайн-сервис (рисунок 3).

Изображение выглядит как текст, электроника, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 3 – Пример работы сервиса по анализу текста

На рисунке 3 показана работа сервиса с текстовым файлом, в котором находился текст сочинения. Сервис отформатировал документ, оформив текст в нем в соответствие с ГОСТом, исправив стили заголовков, проставив отступы и исправив типографические ошибки.

Рассматриваемый сервис автоматически форматирует учебные работы, загруженные с расширением docx в соответствии с ГОСТами, исправляя шрифт, отступы, изменяя заголовки и другие ошибки.

Вторым примером рассмотрим еще один онлайн-сервис (рисунок 4).

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Автоматически созданное описание

Рисунок 4 – Пример сервиса для поиска и проверки стандартов

Рассматриваемый сервис позволяет проверить на соответствие не только ГОСТу, но и СНиП, СанПиН, РД и СП. В приведенном сервисе можно загружать файлы расширения: doc/docx, pdf, xls /xlsx.

Для работы в сервисе нужно указать наименование стандарта или регламента, затем загрузить файл для проверки, будет производиться поиск и проверка в документе, указанного стандарта и регламента.

Сервис предоставляет две версии: платную и демоверсию. Также сервис располагает многими другими возможностями.

**3 Проектирование сервиса определения соответствия технической документации по ГОСТу**

Процесс написания технической документации заключается в описании и сопровождении программного продукта в течение их жизненного цикла.

Проверка технической документации:

1. Загрузка технической документации

2. Загрузка шаблонов (Приложение А)

3. Сравнение технической документации с шаблоном

4. Проверки в соответствии с типом документации

5. Проверка характеристик шрифта

6. Проверка межстрочного интервала и отступов в начале абзаце, выравнивания текста по ширине

7. Формирование отчёта о несходствах с ГОСТом

Действующие лица:

Технический эксперт

Процесс написания технической документации заключается в описании и сопровождении программного продукта в течение их жизненного цикла.

Входная и выходная информация, структурная схема типа IDEF0 представлена на рисунке 5.



Рисунок 5 – Модель сервиса

Ниже на рисунке 6 приведена диаграмма декомпозиции А0.

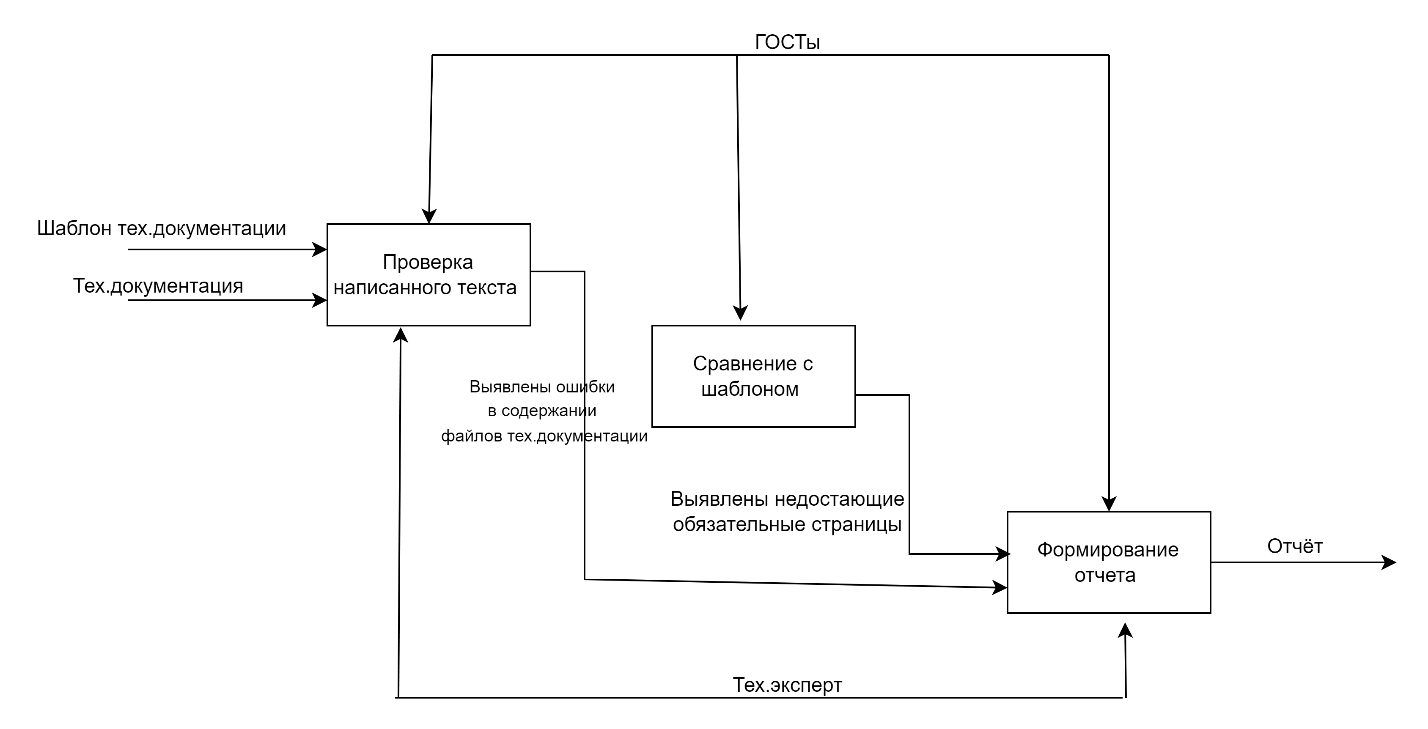


Рисунок 6 – Диаграмма декомпозиции А0

Далее приведем диаграмму прецедентов на рисунке 7.

Изображение выглядит как текст, снимок экрана, круг, луна

Автоматически созданное описание

Рисунок 7 – UML-диаграмма прецедентов

Разрабатываемый сервис должен решать такие задачи:

* задача корректного составления структуры шаблона (Приложение А),
* задача проверки содержания технической документации,
* задача проверки структуры,
* задача проверки характеристик шрифта,
* задача проверки составляющих содержания,
* задача выделения ошибок,
* задача визуализации результата.

Требования к задаче «Задача корректного составления структуры шаблона»

1. Для создания шаблонов для существующих видов технической документации нужно изучить ГОСТы, в которых указаны особенности ее создания. К таким ГОСТам относится например ГОСТ 19.001–77, 19.201–78, 19.105–78

2. Шаблон должен содержать все обязательные страницы технической документации (Приложение А).

Требования к задаче «Задача проверки содержания технической документации»

1. Техническая документация должна иметь обязательные страницы.

2. У разных типов технической документации есть свои обязательные пункты, описанные в конкретных ГОСТах, например, количество строк кода в типе технической документации «Текст программы».

Требования к задаче «Задача проверки структуры»

1. В технической документации текст должен быть оформлен в соответствии с ГОСТом.

2. По установленным ГОСТом правилам текст в технической документации должен быть выровнен по ширине, отступы в начале абзаца должен быть 1.25, межстрочный интервал 1.5.

Требования к задаче «Задача проверки характеристик шрифта»

1. В ГОСТах установлен стиль и размер шрифта для всех типов технической документации, это соответственно Times New Roman и 14.

Требования к задаче «Задача проверки составляющих содержания»

1. Написанный текст в содержании должен соответствовать ГОСТу 19.402–78 ЕСПД Описание программы, 19.401–78 ЕСПД Текст программы. Требования к содержанию и оформлению, содержать обязательные пункты и подпункты.

Требования к задаче «Задача выделения ошибок»

1. Найденные ошибки в содержании, нехватка обязательных страниц и прочие несоответствия должны быть выделены.

Требования к задаче «Задача визуализации результата»

1. В консоли написанного сервиса должны быть выведены ошибки, найденные в загруженном файле технической документации.

Ниже на рисунке 8 приведена диаграмма требований (вариантов использования) UML отражающая основной функционал сервиса.

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок 8 – Модель требований в нотации UML

Проведем декомпозицию на подзадачи.

Ниже на рисунке 9 представлена декомпозиция «Проверка написанного текста».



Рисунок 9 – Декомпозиция «Проверка написанного текста»

Далее приведем диаграмма состояний (statechart diagram), которая описывает состояние проверки технической документации на протяжении всего цикла работы сервиса (рисунок 10).

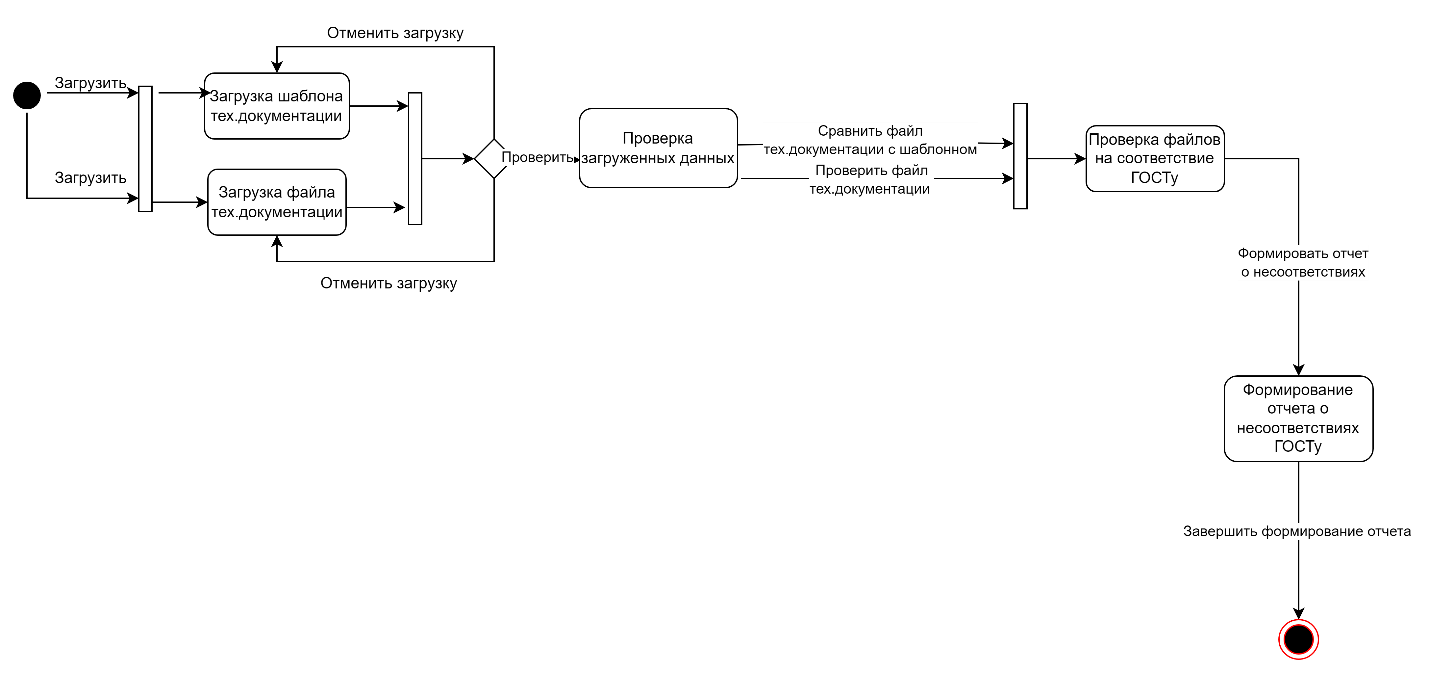


Рисунок 10 – Диаграмма состояний

Ниже приведем диаграмму активности (activity diagram). Это диаграмма, описывающая процесс проверки технической документации (рисунок 11).

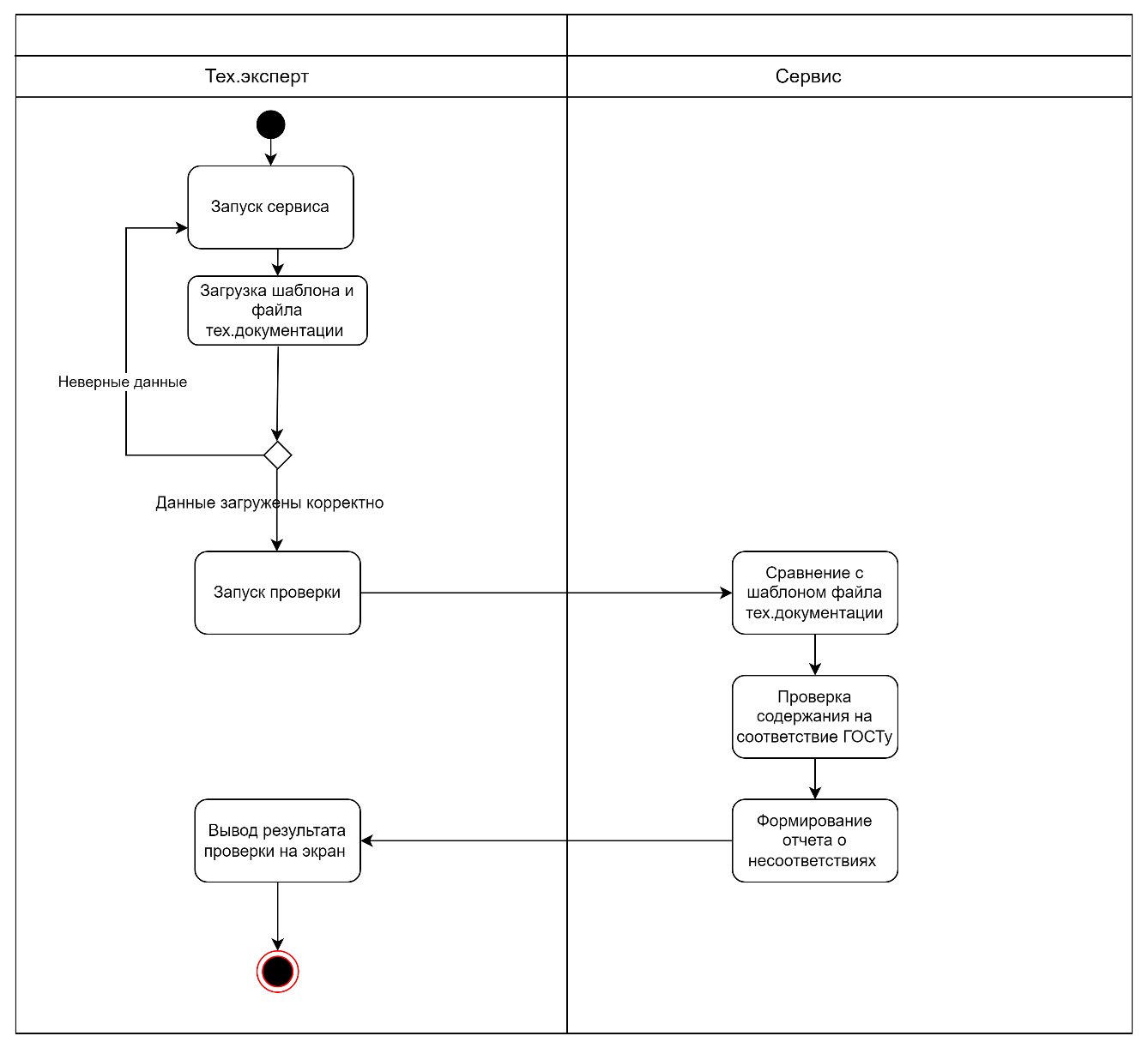


Рисунок 11 – Диаграмма активности

Далее приведена диаграмма последовательностей (sequence diagram), которая описывает жизненный цикл объекта в рамках прецедента (рисунок 12).

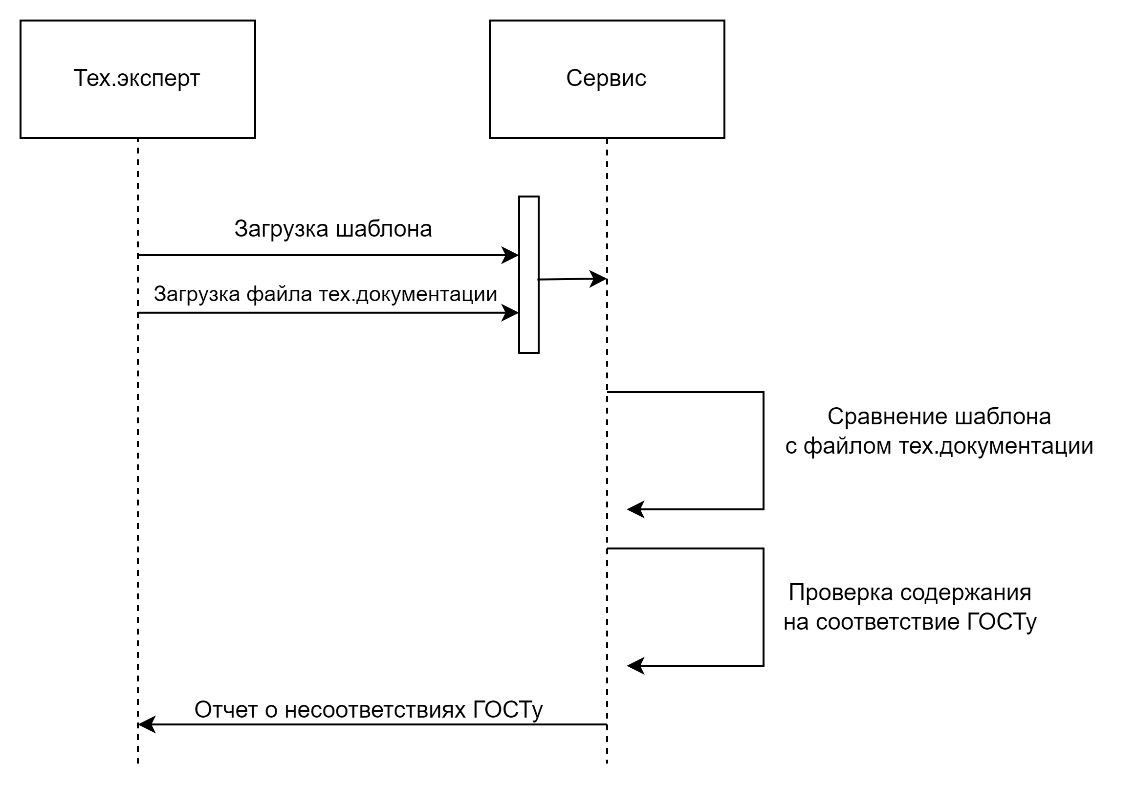


Рисунок 12 – Диаграмма последовательностей

1. **Особенности реализации сервиса и примеры работы**

Информация о производстве, реализации, обслуживание и использовании описывается в программной документации.

Технической документацией являются такие файлы как: «Описание программы», «Текст программы», «Руководство системного программиста», «Описание применения» и «Руководство оператора» [8].

Чтобы определить особенности каждого типа документации и общепринятых характеристик, и чтобы разработать сервис, были проанализированы ГОСТы. ГОСТы подразделяются на, те, в которых описаны общие требования для всех видов, и те, которые относятся конкретно к какому-либо виду технической документации. При их изучении были выделены особенности:

* Некоторые из видов технической документации, такие как: «Описание программы», «Описание применения» и другие, имеют строго регламентированные пункты и подпункты содержания, что, в частности, указанно в соответствующих каждому типу ГОСТах.
* Во всех документах межстрочный интервал 1.5, а отступ в начале абзаца 1.25, текст выровнен по ширине.
* Техническая документация, в которой стиль шрифта текста должен быть ‘Times New Roman’, а размер 14.

Для реализации проекта использован язык Python с подключением кроссплатформенной событийно-ориентированной графической библиотеки tkinter, также библиотеки pathlib, docx, docx2txt. Программа содержит пять функций, работающих с файлами.

Алгоритм проверки соответствия технической документации ГОСТу основан на файл-шаблоне, который составлен техническим экспертом и содержит все закрепленные части, за исключением основной. В шаблоне присутствуют вариативные поля, являющиеся индивидуальными для каждого проекта, к ним относятся, например, название программы, заголовки и необходимые обозначения, пункты содержания. Файлы-шаблоны соответствуют каждому типу технической документации и могут быть загружены с локального компьютера или с любого внешнего носителя. Файлы-шаблоны представлены в формате docx, как и большинство стандартных файлов технической документации.

При работе сервиса с любым из типов технической документации, тестируемый файл документации и файл-шаблон сопоставляются с помощью методов библиотек docx и docx2txt языка программирования Python. Выявляются первичные недостатки, заключающиеся в отсутствии закрепленных страниц. Результатом первого этапа обработки является вывод в текстовое поле интерфейса количества обязательных страниц, соответствующих шаблонным (Приложение А).

Второй этап обработки тестируемого файла выполняется в соответствии с типом документа, так как кроме общих требований к технической документации, существуют специфические нормативы, отражающие особенности документа. Так, например ГОСТ 19.401–78, регулирует структуру и содержание файла конкретного типа, в частности «Текст программы» [8]. На этом этапе контролируются конкретные требования к каждому типу технической документации, проводятся индивидуальные тесты соответствия. При проверке файла технической документации «Текст программы» помимо первичной обработки ГОСТом, устанавливается количество строк кода, которое должно быть не менее 500. Сервис подсчитывает количество строк, причем для точности подсчета, в файле технической документации не рассматриваются первые три страницы, которые не содержат кода и являются закрепленными.

При проверке файла технической документации «Описание программы» контролируются определенные ГОСТом 19.402–78 обязательные пункты и подпункты содержания [8]. В этом случае сервис выделяет ключевые слова, выполняя токенизацию текста документации и сравнивая найденные слова с закрепленными. На первом этапе алгоритма методами библиотеки docx2txt выполняется извлечение слов из текста, приведение их к нижнему регистру. На следующем этапе реализуется сравнение слов с ключевыми словами, требуемыми по ГОСТу и хранимыми в массиве закрепленных слов. При полном совпадении всех пунктов проверки в текстовое поле интерфейса будет выведено сообщение о соответствии содержания ГОСТу, в противном случае – перечень выявленных несоответствий.

В сервис добавлена возможность проверки файлов «Руководство оператора», «Руководство системного программиста», «Описание применения», а в частности проверка пунктов и подпунктов содержания.

Фрагмент функции, сравнивающей исходный файл с соответствующим файлом-шаблоном, содержащим обязательные по ГОСТу страницы.def content\_op():

text = docx2txt.process(file1\_entry.get())

f = Path(file2\_entry.get()).stem

text = text.lower()

if f == 'ШаблонОписаниепрограммы':

search\_words = ['общие сведения', 'функциональное назначение', 'описание логической структуры',

'используемые технические средства', 'вызов и загрузка', 'входные данные', 'выходные данные']

for search\_word in search\_words:

if search\_word.lower() not in text:

return result\_text.insert("5.0", "\n Cодержаниe файла 'Описание программы' не соответсвует ГОСТу: " + "\n\n")

return result\_text.insert("5.0", "\nCодержаниe файла 'Описание программы' соответсвует ГОСТу: " + "\n\n")

Функция content\_op() проверяет какой шаблон загружен, шаблон уникален для каждого типа документации, также как и пункты и подпункты содержания, после определения соответствия между шаблоном и технической документацией, файл проверяется на наличие ключевых слова и словосочетания, из которых формируются разделы .

Так для проверки шрифта написана функция.

def check\_font():

doc1 = Document(file1\_entry.get())

doc2 = docx.Document(file2\_entry.get())

b = (len(file1\_entry.get()) - 4) \* 2

text = docx2txt.process(file1\_entry.get())

for paragraph in doc1.paragraphs:

for run in paragraph.runs:

if run.font.name != "Times New Roman" or run.font.size != 14:

return result\_text.insert("3.0", "\n Текст не соответствует по шрифту ГОСТу " + "\n\n")

return result\_text.insert("3.0", "\nТекст соответствует по шрифту ГОСТу " + "\n\n")

Функция check\_font() отбрасывает обязательные страницы, для них необходима проверка детальнее, после остальные страницы, которые составляют основную часть в каждом виде документации, просматриваются на присутствие в ней текста, который должен соответствовать размеру 14, а по стил шрифта Times New Roman, если текст удовлетворяет заданные условия одновременно, то на экране будет показана информация о том, что стиль шрифта и размер в технической документации соответствует ГОСТу.

Для проверки различных особенностей текста: межстрочный интервал, отступ в начале абзаца, выравнивание по ширине, написана функция.

def check\_all():

doc1 = Document(file1\_entry.get())

doc2 = docx.Document(file2\_entry.get())

b = (len(file1\_entry.get()) - 4) \* 2

text = docx2txt.process(file1\_entry.get())

line\_spacing\_ok = True

for paragraph in doc1.paragraphs:

if paragraph.paragraph\_format.line\_spacing != 1.5:

line\_spacing\_ok = False

break

if line\_spacing\_ok:

result\_text.insert("3.0", "\nТекст соответствует по межстрочному интервалу ГОСТу " + "\n\n")

else:

result\_text.insert("3.0", "\n Текст не соответствует по межстрочному интервалу ГОСТу " + "\n\n")

indentation\_ok = True

for paragraph in doc1.paragraphs:

if paragraph.paragraph\_format.left\_indent != 1.25:

indentation\_ok = False

break

if indentation\_ok:

result\_text.insert("3.0", "\nТекст соответствует по отступу в начале абзаца ГОСТу " + "\n\n")

else:

result\_text.insert("3.0", "\n Текст не соответствует по отступу в начале абзаца ГОСТу " + "\n\n")

alignment\_ok = True

for paragraph in doc1.paragraphs:

if paragraph.paragraph\_format.alignment != WD\_ALIGN\_PARAGRAPH.JUSTIFY:

alignment\_ok = False

break

if alignment\_ok:

result\_text.insert("3.0", "\nТекст соответствует по выравниванию ГОСТУ " + "\n\n")

else:

result\_text.insert("3.0", "\n Текст не соответствует по выравниванию ГОСТу " + "\n\n")

Функция def check\_all() также отбрасывают обязательные страницы, а затем проверяют текст чтобы он выровнен по ширине файла, отступ в начале абзаца равен 1.25, а межстрочный интервал равен 1.5.

В сервисе еще реализованы несколько функций для проверки ошибок, такие как

* функция content\_op()

примером является функция, сравнивающая исходный файл с соответствующим файлом-шаблоном, содержащим обязательные по ГОСТу страницы.

def compare\_files():

file1\_path = Document(file1\_entry.get())

file2\_path = Document(file2\_entry.get())

template\_pages = len(file2\_path.sections)

count = 0

same\_pages = []

for i in range(len(file1\_path.sections)):

for j in range(len(file2\_path.sections)):

if file1\_path.paragraphs[i].text == file2\_path.paragraphs[j].text:

same\_pages.append(i)

count += 1

result\_text.delete("0.0", tk.END)

result\_text.insert("1.0", "Одинаковые страницы: " + str(

same\_pages) + "\nКоличество одинаковых страниц: " + str(count) + "\n\n")

Функция compare\_files() разделяет загруженные файлы на секции, затем сравнивает их и выводит одинаковые секции, а также количество одинаковых страниц.

* функция n\_o\_r ()

Примером является функция, считающая количество строк кода в файле.

def n\_o\_r():

doc1 = docx.Document(file1\_entry.get())

f = Path(file2\_entry.get()).stem

if f == 'ШаблонТекстпрограммы':

line\_count = 0

for paragraph in doc1.paragraphs[3:-1]:

line\_count += len(paragraph.text.splitlines())

result\_text.delete("3.0", tk.END)

result\_text.insert("3.0", "\nКоличество строк: " + str(line\_count) + "\n\n")

Функция n\_o\_r() проверяет какой шаблон был загружен, в случае, если это шаблон для файла «Текст программы», то не учитываются обязательные страницы: они не содержат кода. Затем вычисляется количество строк на оставшихся страницах и результат выводится на экран.

Рассмотрим работу сервиса на конкретных примерах.

Пример 1.

Ниже приведен фрагмент файла технической документации «Описание применения». На рисунке 13 видно, что в тексте есть отличия по межстрочному интервалу, отступу в начале абзаца, в остальном текст выровнен по ширине, характеристики шрифта соответствует установленным ГОСТу.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 13 – Фрагмент технической документации «Описание применения»

Ниже на рисунке 14 фрагмент шаблона, на котором представлено содержание, с которым не только происходит сравнение для выявления наличия обязательной страница, но и проверка содержания на наличие все обязательных пунктов.

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок 14 – Фрагмент шаблона технической документации «Описание применения»

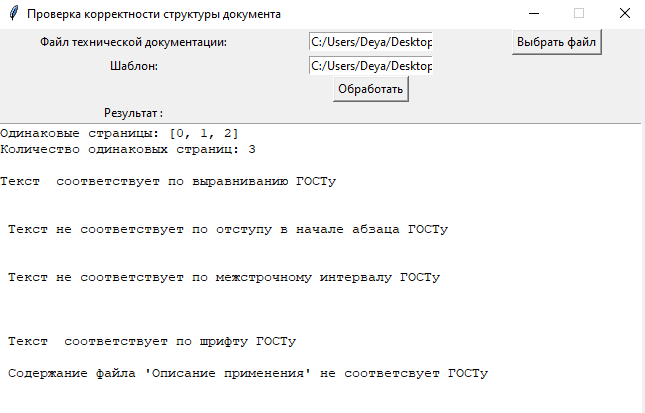


Рисунок 15 – Пример работы сервиса с техническим документом «Описание применения»

На рисунке 15 можно увидеть работу сервиса с техническим документом «Описание применения».

В примере 1 можно увидеть, что проведено сравнение с шаблоном и выведены одинаковые страницы, их количество, выполнена проверка содержания, текст проверен на соответствие типу шрифта и размеру, проверены отступы в начале абзацев и межстрочный интервал.

Пример 2.

Ниже приведен фрагмент файла технической документации «Руководство Оператора». На рисунке 16 видно, что в текст правильно оформлен: межстрочный интервал, выравнен по ширине, отступы в начале абзаца, характеристики шрифта соответствует установленным ГОСТу.

Изображение выглядит как текст, снимок экрана, Шрифт, письмо

Автоматически созданное описание

Рисунок 16 – Фрагмент технической документации «Руководство оператора»

Ниже на рисунке 17 фрагмент шаблона, на котором представлен лист аннотации, с которым не только происходит сравнение для выявления наличия обязательной страница.

Изображение выглядит как снимок экрана, белый, текст, дизайн

Автоматически созданное описание

Рисунок 17 – Фрагмент шаблона технической документации «Руководство оператора»

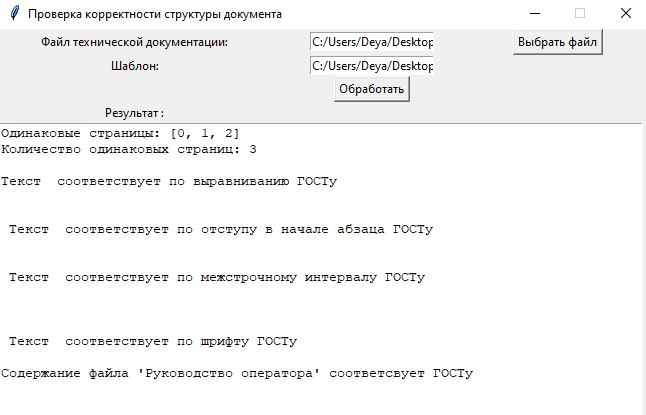


Рисунок 18 – Пример работы сервиса с техническим документом «Руководство оператора»

На рисунке 18 можно увидеть работу сервиса с техническим документом «Руководство оператора».

В примере 2 можно увидеть, что проведено сравнение с шаблоном и выведены одинаковые страницы и их количество, также проведена проверка содержания на соответствие ГОСТу и произведена проверках всех установленных пунктов для шрифта, межстрочный интервал и отступы в начале абзаца.

**ЗАКЛЮЧЕНИЕ**

Цель выпускной квалификационной работы – проектирование и реализация сервиса определения соответствия технической документации ГОСТу – достигнута.

В теоретической части выпускной квалификационной работы описаны технологии обработки естественного языка, а также выделены базовые задачи анализа текста.

Выполнен реферативный обзор сервисов, реализующих проверку соответствия ГОСТу. Спроектирован сервис определения соответствия технической документации по ГОСТу.

В работе предложен проект и реализация сервис определения соответствия технической документации по ГОСТу. Основной функцией сервиса является поиск несоответствий с ГОСТами в структуре документов.

При программной реализации функционала использовалась библиотека языка Python tkinter. При работе сервис находит такие ошибки, например, как отсутствие обязательных страниц, недостаточное количество строк кода и несоответствие содержания ГОСТу, неверный тип или размер шрифта, неверные отступы в начале абзацев и неверно выравненный текст.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Автоматическая обработка текстов на естественном языке и компьютерная лингвистика: учеб. пособие / Большакова Е.И., Клышинский Э.С., Ландэ Д.В., Носков А.А., Пескова О.В., Ягунова Е.В. — М.: МИЭМ, 2011. — 272 с.

2. Беляев, И.Н. Технологии обработки текстов: учебное пособие / И.Н. Беляев. – СПб.: БХВ-Петербург, 2015. – 280 с.

3. Автоматическая обработка текстов на естественном языке и анализ данных: учеб. пособие / Большакова Е. И., Воронцов К. В., Ефремова Н.Э., Клышинский Э.С, Лукашевич Н.В., Сапин А.С – М.: Изд-во НИУ ВШЭ, 2017. – 269 с.

4. Агеев, М. С. Автоматическая рубрикация текстов: методы и проблемы / М. С. Агеев., Б. В. Добров, Н. В. Лукашевич– Казань: Физико-математические науки, — 2008.— Т. 150, № 4.— С. 25–40.

5. Гайсин, А.Э. Анализ существующих методов автоматического текстового анализа // Вестник науки. 2023. №6 (63). URL: <https://cyberleninka.ru/article/n/analiz-suschestvuyuschih-metodov-avtomaticheskogo-tekstovogo-analiza> (дата обращения: 17.10.2023).

6. Анализ текста и его обработка: [сайт]. – 2022 – URL: <https://nitforyou.com/avtomaticheskaja-obrabotka-teksta/> (дата обращения: 17.10.2023).

7. Митина, О.В, Евдокименко, А.С. Формализованные методы исследования текстов: опыт применения к анализу технической документации // Вестн. Том. гос. ун-та. Филология. 2010. №1 (9). URL: <https://cyberleninka.ru/article/n/formalizovannye-metody-issledovaniya-tekstov-opyt-primeneniya-k-analizu-tehnicheskoy-dokumentatsii> (дата обращения: 17.11.2023).

8. ГОСТ 19.105–78, ГОСТ 19.402–78, ГОСТ 19.101–77, ГОСТ 19.104–78. Единая система программной документации. Общие требования к программным документам: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3350: введен впервые: дата введения 1980-01-01.

**ПРИЛОЖЕНИЕ**

**Шаблон технической документации «Описание применения»**Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Рисунок А.1 – Шаблон технической документации «Описание применения»

Изображение выглядит как текст, снимок экрана, белый, дизайн

Автоматически созданное описание

Рисунок А.2 – Вторая страница шаблона технической документации «Описание применения»

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок А.3 – Третья страница шаблона технической документации «Описание применения»

Изображение выглядит как текст, снимок экрана, Параллельный, линия

Автоматически созданное описание

Рисунок А.4 – Четвертая страница шаблона технической документации «Описание применения»

**Шаблон технической документации «Руководство оператора»**

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок А.5 – Первая страница шаблона технической документации «Руководство оператора»

Изображение выглядит как снимок экрана, белый, дизайн, алгебра

Автоматически созданное описание

Рисунок А.6 – Шаблон технической документации «Руководство оператора» вторая страница

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок А.7 – Шаблона технической документации «Руководство оператора» третья страница

Изображение выглядит как текст, линия, диаграмма, Параллельный

Автоматически созданное описание

Рисунок А.8 – Шаблон технической документации «Руководство оператора» лист регистрации

**Шаблон технической документации «Текст программы»**

Изображение выглядит как текст, снимок экрана, диаграмма, линия

Автоматически созданное описание

Рисунок А.9 – Шаблона технической документации «Текст программы» титульный лист

Изображение выглядит как снимок экрана, текст, белый, дизайн

Автоматически созданное описание

Рисунок А.10 – Шаблон технической документации «Текст программы» аннотация

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

Рисунок А.11 – Шаблон технической документации «Текст программы» содержание

Изображение выглядит как текст, линия, Параллельный, чек

Автоматически созданное описание

Рисунок А.12 – Шаблон технической документации «Текст программы» четвертая страница