ВВЕДЕНИЕ

Люди и компании часто сталкиваются с необходимостью обработки конфиденциальных документов, содержащих персональные данные. При этом требуется соблюдать баланс между сохранением полезной информации и защитой приватности данных. Традиционные методы анонимизации документов, основанные на ручной обработке, требуют значительных временных затрат и подвержены человеческим ошибкам.

Современные технологии обработки естественного языка открывают новые возможности для автоматизации процесса анонимизации, позволяя существенно сократить время обработки документов и повысить качество результата. Особенно актуальной эта задача становится в условиях ужесточения законодательства о защите персональных данных и увеличения объемов обрабатываемой информации.

Целью данной работы является разработка автономного сервиса для автоматической анонимизации персональных данных в текстовых документах на русском языке. Система реализует гибридный подход, сочетающий традиционные алгоритмы и современную языковую модель для контекстного анализа. Особое внимание уделено возможности добавления пользовательских типов сущностей без необходимости обучения моделей.

Разрабатываемый сервис предоставляет удобный веб-интерфейс для загрузки и обработки документов, а также API для интеграции с существующими корпоративными системами. Важной особенностью решения является возможность локального развертывания, что обеспечивает высокий уровень безопасности при работе с конфиденциальными данными. Гибкая система настроек позволяет адаптировать процесс анонимизации под специфические требования различных организаций и типов документов.

1 ГЛАВА 1. ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ И СУЩЕСТВУЮЩИЕ РЕШЕНИЯ

1.1 Концепция анонимизации персональных данных

В современном информационном обществе вопросы защиты и конфиденциальности персональных данных приобретают всё большую актуальность. Стремительный рост объёмов обрабатываемой информации и интенсивный обмен данными между различными системами создают предпосылки для поиска эффективных методов обеспечения конфиденциальности сведений о физических лицах.

Согласно Федеральному закону №152-ФЗ «О персональных данных» [3], персональные данные определяются как *«любая информация, относящаяся к прямо или косвенно определенному или определяемому физическому лицу (субъекту персональных данных)»*. Это определение охватывает чрезвычайно широкий спектр сведений, которые могут содержаться в текстовых документах – от прямых идентификаторов личности до косвенных признаков, которые в совокупности могут позволить идентифицировать субъекта данных.

В том же законе[3] даётся определение процесса обезличивания персональных данных как *«действия, в результате которых становится невозможным без использования дополнительной информации определить принадлежность персональных данных конкретному субъекту персональных данных»*. Данное определение закладывает основу для понимания сущности анонимизации как процесса, который не обязательно подразумевает полное уничтожение персональной информации, но делает невозможной идентификацию субъекта без привлечения дополнительных сведений.

Важно отметить, что в контексте обработки персональных данных законодательство вводит понятие оператора, определяемого как *«государственный орган, муниципальный орган, юридическое или физическое лицо, самостоятельно или совместно с другими лицами организующие и (или) осуществляющие обработку персональных данных, а также определяющие цели обработки персональных данных, состав персональных данных, подлежащих обработке, действия (операции), совершаемые с персональными данными»*. Именно на оператора возлагается ответственность за обеспечение конфиденциальности обрабатываемых данных, включая их надлежащую анонимизацию.

Для полного понимания концепции анонимизации необходимо также учитывать определение уничтожения персональных данных. Федеральный закон №152-ФЗ[3] определяет этот процесс как *«действия, в результате которых становится невозможным восстановить содержание персональных данных в информационной системе персональных данных и (или) в результате которых уничтожаются материальные носители персональных данных»*. В отличие от уничтожения, при котором данные перестают существовать в принципе, анонимизация сохраняет данные в измененном виде, при котором их информационная ценность может быть сохранена, но идентификация конкретного субъекта становится невозможной без дополнительной информации.

В контексте текстовых документов анонимизация представляет особую сложность,

поскольку персональные данные могут быть представлены в разнообразных формах и контекстах, что требует применения комплексных подходов для их выявления и обезличивания при сохранении общего смысла и структуры документа.

1.2 Актуальность автоматической анонимизации персональных данных

Организации вынуждены уделять значительное внимание процессам анонимизации информации для соблюдения требований законодательства и защиты конфиденциальности субъектов данных. При этом объёмы обрабатываемых текстовых документов, содержащих персональные данные, непрерывно растут.

Традиционный процесс анонимизации персональных данных в документах обычно включает два последовательных этапа: на первом этапе один сотрудник выполняет непосредственную обработку текста, выявляя и маскируя персональные данные, а на втором этапе другой сотрудник осуществляет проверку и корректировку результатов. Такой двухэтапный подход, хотя и снижает вероятность ошибок, является крайне трудоёмким и времязатратным, что существенно увеличивает стоимость обработки документации.

Ручная анонимизация неизбежно сопряжена с человеческим фактором – усталостью, невнимательностью, субъективностью в интерпретации данных. Как следствие, возникают ошибки, которые имеют серьезные последствия – от нарушения конфиденциальности до утраты аналитической ценности документов.

Особую актуальность проблема приобретает в контексте жёстких санкций за нарушение законодательства о персональных данных. Согласно статье 13.11 КоАП РФ[1], нарушение требований по обработке персональных данных влечет административную ответственность, причем за повторное нарушение требований по хранению персональных данных граждан РФ штрафы увеличиваются вплоть до 18 миллионов рублей для юридических лиц. В отдельных случаях нарушения могут квалифицироваться по статье 137 УК РФ[2] как незаконное собирание или распространение сведений о частной жизни, что предусматривает уголовную ответственность.

Автоматизация процесса анонимизации позволяет существенно снизить трудозатраты, заменив первый этап обработки документов программным решением. При этом сохраняется возможность человеческой проверки и корректировки результатов на втором этапе, что обеспечивает необходимый уровень качества без значительных временных затрат. Это особенно важно для организаций, регулярно работающих с большими объемами документов, содержащих персональные данные – медицинских учреждений, государственных органов, финансовых организаций.

1.3 Описание существующих решений

1.3.1 Microsoft Presidio

Microsoft Presidio[13] представляет собой SDK с открытым исходным кодом для обнаружения и анонимизации персональных данных в текстовых документах. Разработан корпорацией Microsoft и предоставляется на условиях лицензии MIT. Основой технологии служат предобученные модели (трансформерами, такими как например BERT[6]) из библиотеки spaCy[10].

К плюсам данного решения можно отнести следующее:

- Анонимизация изображений
- Поддержка различных методов анонимизации (маскирование, шифрование, удаление)
- Простая интеграция
- Открытый исходный код и активное сообщество разработчиков
- Возможность локального развёртывания без подключения к внешним сервисам

Также у решения существуют и недостатки:

- Отсутствие встроенной поддержки русского языка
- Требуется поиск моделей, подходящего формата, или дообучение для работы с русскоязычными текстами
- Работает только с обычным текстом, нет встроенной поддержки структурированных форматов
- Невозможно добавить свои сущности для их определения без дообучения модели
- Решение требует технической экспертизы для адаптации под специфические требования

1.3.2 Lingvanex

Lingvanex[12] — коммерческий сервис, предоставляющий решения для анонимизации персональных данных в документах различных форматов. Компания специализируется на обработке текстов на различных языках, включая русский. Но при этом решение возможно развернуть локально.

К плюсам решения можно отнести:

- Встроенная поддержка русского языка
- Работа с множеством форматов документов (PDF, DOC, TXT, RTF и др.)
- Наличие готового программного обеспечения с пользовательским интерфейсом

- Возможность локального развёртывания

Однако у решения также существует ряд минусов:

- Коммерческое решение с годовой подпиской
- Закрытый исходный код ограничивает возможности расширения
- В свободном доступен нет информации о наличии кастомизации
- Ограниченная прозрачность технической реализации

1.3.3 Sber DeepDocs

Sber DeepDocs[15] — часть экосистемы Сбера, предоставляющая API для анонимизации документов.

К плюсам данного решения относятся:

- Поддержка русского языка
- Работа с различными форматами документов (PDF, DOC, TXT, RTF и др.)
- Техническое решение от крупной компании
- Все вычислительные затраты ложатся на плечи Сбера

К недостаткам решения можно отнести:

- Коммерческое решение с тарифными планами
- Только облачная реализация, отсутствие возможности локального развёртывания
- В свободном доступен нет информации о наличии кастомизации
- Зависимость от внешней инфраструктуры

1.3.4 Anonympy

Anonympy[4] — библиотека с открытым исходным кодом для анонимизации персональных данных, преимущественно ориентированная на работу с данными в формате PDF. Этот пакет был разработан таким образом, чтобы быть максимально интуитивно понятным

К плюсам решения относятся:

- Простота использования
- Анонимизация изображений
- Открытый исходный код (лицензия BSD)
- Возможность локального развёртывания

К недостаткам решения можно отнести:

- Отсутствие встроенной поддержки русского языка
- Ограниченная поддержка форматов документов (только PDF)
- Нет возможности использовать свои модели

1.3.5 Datanymizer

Datanymizer[5] — инструмент с открытым исходным кодом для анонимизации данных в SQL базах данных. Основан на использовании шаблонов замены без применения машинного обучения, что обеспечивает высокую производительность.

К плюсам данного решения можно отнести:

- Открытый исходный код (лицензия MIT)
- Высокая скорость работы благодаря отсутствию затрат на ML-модели
- Возможность локального развёртывания

Также у решения существуют следующие недостатки:

- Работа только со структурированными данными в базах данных
- Отсутствие поддержки русского языка
- Не предназначен для обработки текстовых документов

1.4 Сравнительный анализ существующих решений

Для обозреваемых конкурентов был проведен сравнительный анализ с разрабатываемым проектом по следующим функциональным возможностям:

1. **Встроенная поддержка русского языка** — возможность обрабатывать тексты на русском языке без дополнительной настройки и адаптации системы.
2. **Используемые ML-модели** — типы и характеристики моделей машинного обучения, применяемых для распознавания и обработки персональных данных в тексте.
3. **Кастомные сущности без дообучения** — возможность добавлять пользовательские типы сущностей для распознавания без необходимости переобучения основной модели.
4. **Кастомизация** — варианты настройки и адаптации системы под специфические потребности.
5. **Лицензия** — тип лицензии, определяющий правовые условия использования программного обеспечения.

6. **Вид решения** — форма реализации программного продукта.
7. **Локальное развертывание** — возможность установки и использования решения в изолированной среде.
8. **Поддерживаемые типы файлов** — форматы документов, с которыми может работать система.
9. **Стоимость** — ценовая политика и модель монетизации продукта.

Результаты представлены в сравнительной таблицке 1.

Таблица 1: Сравнительный анализ конкурентов

Характеристики	**Разраб. проект**	**MS Presidio**	**Lingvanex**	**Sber DeepDocs**	**Anonympy**	**Datanymizer**
Поддержка русского языка	Да	Только с дообучением	Да	Да	Нет	Нет
ML-модели	Vikhr-Gemma-2B	spaCy-кастом.	Нет данных	Нет данных	Transformers	Нет
Кастомные сущности без переобучения	Да	Нет	Нет	Нет	Нет	Нет
Кастомизация	Конфиг	Конфиг	Нет данных	Список сущностей	Выбор метода	T-шаблоны
Лицензия	GPL-3.0	MIT	Коммер.	Коммер.	BSD	MIT
Вид решения	SDK	SDK	ПО	WEB, API	Библиотека	CLI
Локальное развертывание	Да	Да	Да	Нет	Да	Да
Поддерживаемые форматы	TXT, PDF, DOC	Plain Text	PDF, DOC, TXT и др.	PDF, DOC, TXT и др.	PDF	SQL DB
Стоимость	Бесплатно	Бесплатно	Платная подписка	Платные тарифы	Бесплатно	Бесплатно

1.5 Выводы по главе

В данной главе рассмотрена предметная область и обоснована актуальность разработки решения. Проанализированы нормативно-правовые требования к процессу обезличивания данных, которые определяются Федеральным законом №152-ФЗ "О персональных данных".

Проведён анализ существующих решений для анонимизации персональных данных. Выявлены их ключевые преимущества и недостатки, включая проблемы с поддержкой русского языка, ограничения по работе с различными форматами и отсутствие возможности добавления искомым сущностей для специфических задач без переобучения моделей.

На основе представленной информации можно сделать заключение, что разрабатываемое решение может занять свою нишу в вопросе анонимизации персональных данных в России, соблюдая в себе одновременно простоту использования и огромный простор под настройки для каждой задачи.

2 ГЛАВА 2. ОБЗОР ФУНКЦИОНАЛА

2.1 Функциональные требования

В рамках разработки данного автономного сервиса выделяются следующие функциональные возможности:

1. **Обнаружение персональных данных**
   Система должна автоматически распознавать и классифицировать персональные данные или другие заданные данные в текстовых документах. К таким данным например могут относиться:

   - Имена, фамилии, отчества;
   - Адреса проживания;
   - Номера телефонов, паспортные данные, ИНН и СНИЛС;
   - Даты рождения;
   - Номера кредитных карт;

2. **Анонимизация данных**
   Система должна предоставлять возможность анонимизации обнаруженных данных с использованием различных методов:

   - Полное удаление данных;
   - Замена данных специальными символами (звёздочками);
   - Замена данных по шаблону (например, Иванов Иван Иванович $\rightarrow$ <Person>).

3. **Поддержка различных форматов документов**
   Система должна обеспечивать обработку следующих типов файлов:

   - Текстовые документы (TXT);
   - Документы формата DOCX;
   - PDF-документы.

4. **Гибкая настройка правил обработки**
   Администратор должен иметь возможность настроить правила обработки через конфигурационный файл:

   - Включение или отключение поиска определённых типов сущностей (например, поиск только адресов или телефонов);
   - Добавления словаря слов для удаления;
   - Добавление пользовательских типов данных для поиска;
   - Настройка способов замены для каждого типа данных.

5. **Создание отчёта об изменениях**
   После завершения обработки система должна автоматически генерировать отчёт, содержащий:
   - Список обнаруженных персональных данных;
   - Типы данных и их местоположение в документе;
   - Список сущностей для анонимизации.
6. **Редактирование результатов**
   Пользователь должен иметь возможность вручную корректировать результаты анонимизации перед загрузкой итогового документа.
7. **Управление задачами обработки**
   Система должна сохранять информацию о последних выполненных задачах (task ID).
8. **Минималистичный веб-интерфейс**
   Пользовательский интерфейс должен быть простым и интуитивно понятным, предоставляя следующие функции:
   - Загрузка документов для обработки;
   - Просмотр отчётов и результатов;
   - Редактирование результатов;
   - Скачивание результатов и отчётов.

2.2 Нефункциональные требования

Для разрабатываемого сервиса выделяются следующие нефункциональные требования, которые определяют характеристики системы:

2.2.1 Производительность

- Система должна обрабатывать один документ за время, не превышающее среднее время ручной анонимизации человеком при использовании стандартного сервера GPU с объёмом видеопамяти (VRAM) не менее 8 GB и системной оперативной памяти (RAM) не менее 16 GB. При этом учитываются вычислительные затраты на работу языковой модели Vikhr-Gemma-2B-instruct.
- Система должна обрабатывать документы последовательно без потери данных.

2.2.2 Расширяемость

- Возможность добавления новых типов сущностей и правил анонимизации без необходимости изменения исходного кода.

- Конфигурационные файлы должны позволять администратору гибко настраивать систему под конкретные требования.

2.2.3 Совместимость

- Система должна поддерживать обработку текстовых документов в форматах DOCX, PDF и TXT.
- Все выходные данные о произведённых изменениях должны предоставляться в формате json.

2.2.4 Удобство использования

- Веб-интерфейс должен быть интуитивно понятным и минималистичным, обеспечивая лёгкий доступ к основным функциям.
- Администратор должен иметь возможность легко изменять настройки анонимизации через конфигурационный файл JSON.

2.2.5 Безопасность

- Система должна работать исключительно в локальной сети организации, исключая передачу данных через внешние сети.

2.2.6 Надёжность

- Система должна корректно обрабатывать некорректные или повреждённые входные файлы, возвращая информативные сообщения об ошибках.

2.2.7 Локализация

- Интерфейс системы должен быть полностью локализован на русском языке.
- Алгоритмы обработки текста должны учитывать морфологические особенности русского языка.

2.2.8 Документирование изменений

- Система должна автоматически генерировать отчёты о произведённых изменениях в формате json, включая информацию о типах обнаруженных данных.

2.3 Архитектура решения

В данном подразделе описывается архитектура проекта, в соответствии с функциональными и нефункциональными требованиями, описанными ранее.

2.3.1 Общая концепция системы

Разработанная система представляет собой сервис, работающий в закрытом контуре организации и предназначенный для анонимизации персональных данных в текстовых документах различных форматов. Система реализует гибридный подход к анонимизации, сочетая обычные алгоритмы обработки текста (регулярные выражения, словари) с возможностями современных языковых моделей.

Диаграмма прецедентов, представленная на рисунке 1, отображает основные функциональные возможности системы и взаимодействие различных типов пользователей с системой.

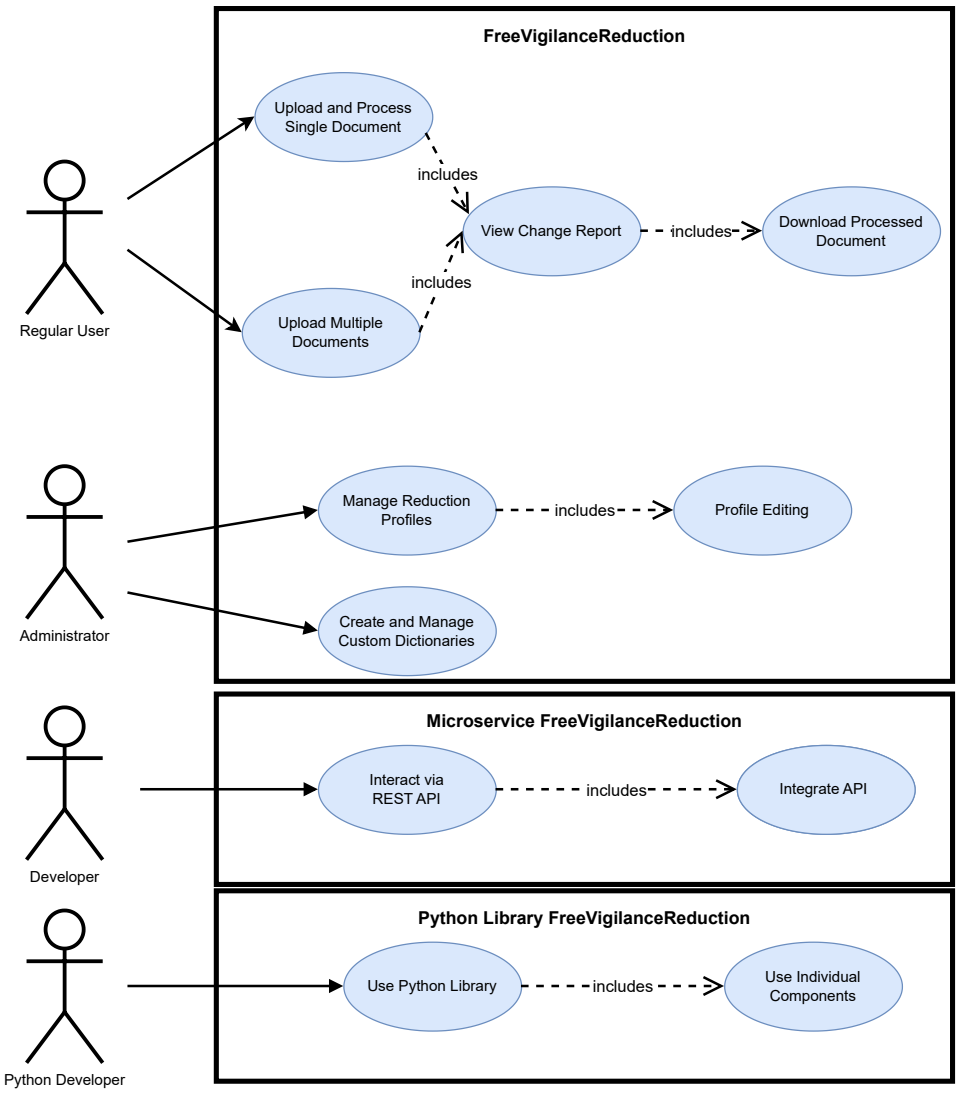

Рис. 1: Диаграмма вариантов использования системы анонимизации

На диаграмме представлены три функциональных блока системы:

- **FreeVigilanceReduction** — блок системы, доступный для обычных пользовате-

лей, который обеспечивает загрузку и обработку документов, просмотр отчетов об изменениях и получение обработанных файлов;

- **Microservice FreeVigilanceReduction** — микросервисный компонент, предоставляющий API для взаимодействия через REST;
- **Python Library FreeVigilanceReduction** — программная библиотека, которая может быть использована разработчиками для интеграции функций анонимизации в собственные проекты, причем как целиком, так и отдельными компонентами.

Диаграмма также определяет четыре типа пользователей:

- **Regular User** — обычный пользователь, работающий с основным блоком системы;
- **Administrator** — администратор, управляющий профилями анонимизации и словарями;
- **Developer** — разработчик, интегрирующий функциональность через API;
- **Python Developer** — разработчик на Python, использующий программную библиотеку.

Такое структурирование системы обеспечивает гибкость её использования в различных сценариях: от непосредственной работы с документами через пользовательский интерфейс до глубокой интеграции функций анонимизации в существующие корпоративные решения.

2.3.2 Взаимодействие компонентов

Процесс взаимодействия пользователей с системой при обработке документов и настройке параметров анонимизации представлен на рисунке 2 в виде BPMN-диаграммы.

Диаграмма демонстрирует два основных потока действий:

- Обработка документов обычным пользователем (верхняя дорожка "Regular User");
- Настройка системы администратором (нижняя дорожка "Administrator").

В потоке обычного пользователя процесс начинается с выбора документов для обработки (работа с одним документом или пакетная обработка). После загрузки документов начинается процесс их обработки, включающий генерацию анонимизированного документа и создание отчета об изменениях. Пользователь может редактировать результаты анонимизации перед итоговым скачиванием обработанных документов.

В административном потоке процесс начинается с редактирования конфигурационного файла. Администратор имеет возможность настраивать три основных компонента:

- Типы сущностей для обнаружения (имена, адреса, телефоны и т.д.);
- Методы замены для каждого типа сущностей;

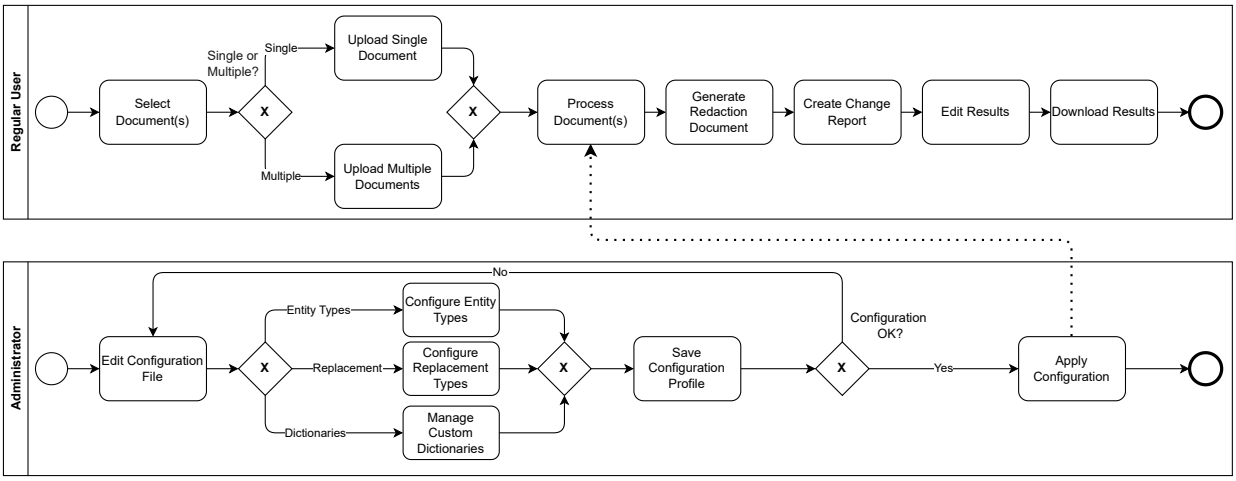

Рис. 2: BPMN-диаграмма процесса анонимизации документа

- Пользовательские словари для улучшения обнаружения специфических данных.

После настройки всех параметров администратор сохраняет конфигурационный профиль. Если настройки удовлетворяют требованиям, администратор применяет конфигурацию, которая немедленно начинает использоваться в процессе обработки документов.

2.4 Описание последовательности обработки документов

Для детального понимания процесса анонимизации документов рассмотрим его последовательность в системе. Диаграмма последовательности, представленная на рисунке 3, иллюстрирует полный жизненный цикл обработки документа: от загрузки файла пользователем до получения результата.

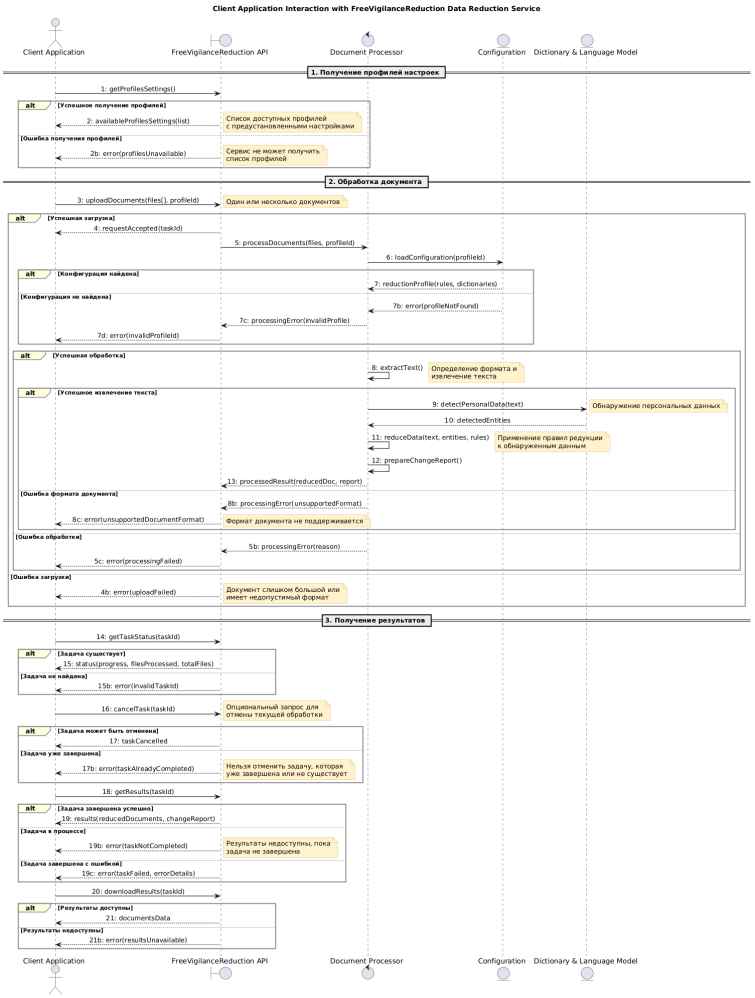

Рис. 3: Диаграмма последовательности процесса анонимизации

2.4.1 Анализ последовательности действий

Для удобства диаграмма последовательности разделена на три основных этапа:

1. **Получение профилей настроек** — взаимодействие начинается с запроса клиентом доступных профилей анонимизации. Этот этап позволяет пользователю выбрать наиболее подходящий набор правил обработки документа в зависимости от его содержания и требований.

2. **Обработка документа** — центральный и наиболее сложный этап, который включает загрузку документа, обработка документа с учётом его формата, обнаружение персональных данных с учётом выбранного профиля, применение правил редукции к обнаруженным данным, подготовку отчета, dозврат обработанных результатов.

3. **Получение результатов** — заключительный этап, на котором можно получить анонимизированный документ и отчет об изменениях.

2.4.2 Взаимодействие компонентов системы

- **Client Application** — клиентское приложение, через которое пользователь взаимодействует с системой. Может быть веб-интерфейсом или внешним приложением, использующим API.
- **FreeVigilanceReduction API** — сервис, обрабатывающий все входящие запросы и управляющий процессом анонимизации.
- **Document Processor** — компонент, отвечающий за обработку документов: извлечение текста, анализ содержимого, применение правил анонимизации и формирование результатов.
- **Configuration** — компонент, управляющий профилями настроек анонимизации.
- **Dictionary & Language Model** — источники данных для обнаружения персональных сведений, а также языковую модель для контекстного анализа.

2.5 Архитектура Python библиотеки

Ядром разработанного решения является Python-библиотека, реализующая основные алгоритмы анонимизации персональных данных. Библиотека спроектирована таким образом, чтобы обеспечить не только использование внутри микросервиса, но и возможность прямого применения без интеграций.

2.5.1 Диаграмма классов библиотеки

Структура библиотеки и взаимосвязи между её компонентами представлены на диаграмме классов (рисунок 6 в приложении А).

Центральным классом библиотеки является `FreeVigilance`, который координирует работу всех компонентов и предоставляет основной интерфейс для взаимодействия с системой.

Библиотека построена с использованием следующих основных компонентов:

1. **Управление конфигурацией** — классы `ConfigurationManager` и `ConfigurationProfile` для загрузки и управления профилями с правилами анонимизации.
2. **Фабрика документов** — `DocumentFactory` и абстрактный класс `Document` с реализациями для конкретных форматов (`DocProcessor`, `PdfProcessor`, `TxtProcessor`).
3. **Распознавание сущностей** — `EntityRecognizer` для обнаружения персональных данных с помощью алгоритмов, `DictionaryManager` и `LanguageModel`.

4. **Замена данных** — `DataReplacer` для применения правил анонимизации к обнаруженным сущностям.

5. **Отчёты об изменениях** — `ReductionReport` с информацией о результатах анонимизации.

6. **Наблюдатели процесса** — интерфейс `ProcessingObserver` для мониторинга этапов обработки документа.

2.6 Алгоритм работы анонимизации

Разработанный сервис анонимизации персональных данных реализует гибридный подход, сочетающий традиционные методы обработки текста с применением языковой модели. Такая комбинация обеспечивает высокую точность обнаружения персональных данных и одновременно с этим предоставляет возможности к кастомизации.

2.6.1 Гибридный подход к обнаружению персональных данных

Трёхуровневый подход к обнаружению персональных данных:

1. **Регулярные выражения** — первый уровень анализа, направленный на обнаружение структурированных данных с фиксированным форматом (номера телефонов, паспортные данные и так далее).

2. **Словарные методы** — второй уровень, использующий предварительно подготовленные списки, куда например организация может внести своё название, имена, фамилии, отчества сотрудников и клиентов.

3. **Анализ с помощью языковой модели** — третий уровень, использующий контекстную информацию для обнаружения персональных данных, которые не были найдены предыдущими методами. Также благодаря наличию языковой модели возможно крайне простое добавление новых сущностей для анонимизации.

2.7 Выбор языковой модели

2.7.1 Почему не используется BERT

Одним из популярных подходов к задаче обнаружения сущностей (Named Entity Recognition, NER) является использование трансформеров типа BERT[6]. На рисунке 4 показана схема работы модели BERT для задачи NER, аналогичной CoNLL-2003[16] (это стандартный датасет для обучения и оценки моделей NER, включающий тексты с размечиными типами сущностей: PER (персона), LOC (локация), ORG (организация) и прочее).

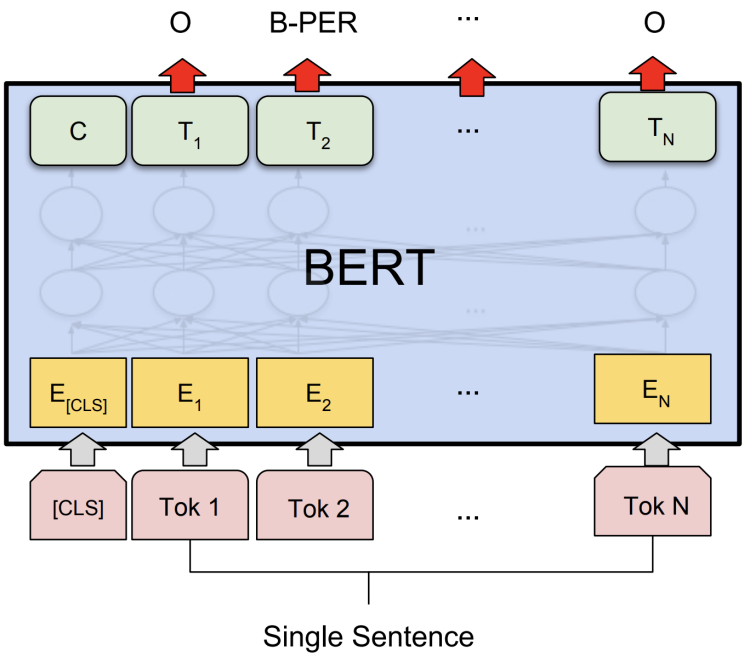

Рис. 4: Использование BERT для задачи NER (на примере CoNLL-2003)

На вход модели BERT подается токенизированное предложение. После прохождения через слои трансформера на выходе формируется векторное представление каждого токена, к которому применяется классификатор, определяющий метку BIO (от англ. Beginning–Inside–Outside). Используется для разметки текста. Метка `B-TYPE` (Begin) указывает на начало сущности определенного типа (например, `B-PER` — начало имени человека), а `I-TYPE` (Inside) — продолжение этой же сущности (например, `I-PER` — продолжение имени), есть также `O` (Outside) — токен, не являющийся частью ни одной сущности. В итоге модель обучается распознавать границы и типы сущностей в тексте, что может являться одним из алгоритмов анонимизации: после определения, какие слова относятся к персональным данным (например, именам, организациям или адресам), система может заменить/удалить их по заданным правилам. Но несмотря на высокую точность таких моделей, их применение имеет ряд ограничений:

- **Зависимость от фиксированного списка сущностей.** BERT и его аналоги ограничены фиксированным числом меток в выходном классификаторе.
- **Необходимость дообучения.** Для поддержки пользовательских сущностей (например, кастомных меток организаций, названий или терминов) требуется дообучать модель на специализированных датасетах, что требует мало того, что значительных ресурсов, так ещё и асессоров для разметки данных.
- **Отсутствие гибкой настройки.** В большинстве BERT-подобных решениях невозможно управлять поведением модели через настройки без вмешательства в архитектуру.

Таким образом, BERT-подобная модель не была выбрана из-за ограничений по адап-

тивности, масштабируемости и удобству в условиях задач промышленной анонимизации.

2.7.2 Переход к архитектуре Generative pre-trained transformer

Появление крупных языковых моделей (LLM, Large Language Models), таких как например семейство GPT[14] от OpenAI, стало важным этапом в развитии обработки языка. Особенно заметным этот сдвиг стал в 2022–2023 годах, когда начался настоящий бум генеративных ИИ. В тот момент модели продемонстрировали способность не только отвечать на вопросы и писать тексты, но и решать прикладные задачи в разных предметных областях.

В отличие от BERT-подобных моделей, которые работают как токен-классификаторы и обрабатывают каждый токен по отдельности, GPT воспринимают текст как целостный контекст. Это позволяет им лучше понимать взаимосвязи между словами и определять сущности на уровне смысла.

Кроме того, такие большие LLM поддерживают принципиально иной способ настройки — промпт-инжиниринг. Вместо переобучения модели для каждой новой задачи, достаточно просто изменить текстовый запрос (промпт). В том числе это делает такие модели особенно удобными для задач анонимизации, где требования к типам данных и стратегиям замены могут меняться в зависимости от организации и контекста.

Таким образом, фокус на генеративных моделях обусловлен как техническими возможностями новых моделей, так и практическими требованиями к гибкости, масштабируемости, которые были указаны ранее.

2.7.3 Критерии выбора модели

При выборе языковой модели для системы анонимизации персональных данных учитывались следующие критерии:

Поддержка русского языка. Для качественной обработки русскоязычных текстов модель должна быть специально обучена на обширном корпусе русскоязычных данных.

Размер и ресурсоэффективность. Модель должна обеспечивать баланс между качеством анализа и требованиями к вычислительным ресурсам.

Способность к логическим рассуждениям. Умение модели анализировать контекст и делать выводы о характере информации на основе окружающего текста.

Лицензия. Модель должна иметь лицензию, допускающую использование в коммерческих проектах и модификацию под конкретные задачи.

Возможность локального развертывания. Критически важным требованием было отсутствие необходимости отправки данных на внешние серверы, что обеспечивает конфиденциальность обрабатываемой информации.

2.7.4 Обоснование использования модели

На основе проведенного анализа для реализации системы была выбрана языковая модель Vikhr-Gemma-2B-instruct[17], которая наилучшим образом соответствует установленным критериям.

Vikhr-Gemma-2B-instruct[17] представляет собой специализированную версию модели Gemma[8] от Google, дообученную на русскоязычном датасете GrandMaster-PRO-MAX (формат вопрос-ответ на русском языке). При этом одним из ключевых её преимуществ является компактный размер модели (всего 2,6 млрд параметров). При этом даже при относительно небольшом размере модель демонстрирует производительность на уровне или даже выше более крупных аналогов, включая GPT-3.5-Turbo и Mixtral 8×7B[11] (для сравнения 12.9 млрд параметров). Модель демонстрирует высокую устойчивость к "галлюцинациям"при работе с русским языком по сравнению с другими моделями аналогичного размера, что критически важно для задачи анонимизации, где ложные срабатывания могут привести как к утечке персональных данных, так и к потере важной информации.

При этом всём модель распространяется под лицензией Apache 2.0, что позволяет свободно использовать и модифицировать её для коммерческих проектов, включая адаптацию к конкретным задачам анонимизации и тонкую настройку под специфические типы документов.

2.8 Выводы по главе

В данной главе было представлено проектирование данной работы. Рассмотрена общая архитектура системы, структура разработанной Python-библиотеки. Представлен алгоритм анонимизации.

Рассмотрение ограничений BERT-подобных подходов позволило обосновать необходимость использования генеративной архитектуры в данной задаче. Также приведено обоснование выбора модели Vikhr-Gemma-2B-instruct, как модели, которая находит требуемый баланс между возможностями, качеством обработки, поддержкой русского языка и размером. Всё это заложило основу для создания итогового решения, способного адаптироваться под разнообразные сценарии анонимизации.

3 ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ

3.1 Реализация ядра библиотеки

Ядро библиотеки FreeVigilanceReduction представляет собой набор взаимосвязанных классов, реализующих основную функциональность. Центральным компонентом является класс FreeVigilanceReduction, который координирует работу всех остальных модулей.

```
class FreeVigilanceReduction:
    def __init__(
        self,
        config_path: Optional[str] = None,
        regex_path: str = "config/regex_patterns.json"
    ):
        ...
        self.config_manager = ConfigurationManager(config_path)
        self.document_factory = DocumentFactory()
        self.entity_recognizer = EntityRecognizer(regex_path)
        self.data_replacer = DataReplacer()
        self.observers: List[ProcessingObserver] = []
        ...

    ...
```

Листинг 1: Инициализация класса FreeVigilanceReduction

При создании объекта класса:

- **Загрузка конфигурации** - создаётся объект ConfigurationManager для управления настройками
- **Инициализация фабрики документов** - создаётся DocumentFactory для работы с документами
- **Загрузка модели** - инициализируется EntityRecognizer с указанным путём к файлу с регулярными выражениями
- **Подготовка замены данных** - создается объект DataReplacer
- **Настройка наблюдателей** - инициализируется список наблюдателей (observers)
- **Регистрация процессоров** - регистрирует стандартные обработчики документов (txt, doc, pdf)

3.2 Обработка документов различных форматов

Одной из ключевых функциональных возможностей библиотеки является обработка документов различных форматов. Для реализации возможности обработки разных

форматов был разработан модуль, включающий абстрактный класс Document и его конкретные реализации для различных форматов. Классическое решения такой задачи, которое уже давно зарекомендовало себя.

3.2.1 Абстрактный класс Document

В основе модуля лежит абстрактный класс `Document`, определяющий общий интерфейс для всех обработчиков документов:

```
class Document(ABC):
    def __init__(self, file_path: str):
        self.file_path: str = file_path
        self.text_content: str = ""
        self.metadata: Dict = {}

    @abstractmethod
    def get_text(self) -> str:
        pass

    @abstractmethod
    def create_redacted_copy(
        self,
        reduced_text: str
    ) -> None:
        pass
    ...
```

Листинг 2: Абстрактный класс Document

Класс определяет два абстрактных метода, которые должны быть реализованы в каждом конкретном обработчике:

- get_text() — извлекает текст из документа
- create_redacted_copy(reduced_text) — создаёт анонимизированную копию документа

3.2.2 Фабрика документов

Реализация паттерна "Фабричный метод". Фабричный метод был выбран для удобного создания обработчиков документов разных форматов через единый интерфейс, что позволяет легко добавлять поддержку новых типов файлов, если это потребуется:

```
class DocumentFactory:
    def __init__(self):
        self.processors: Dict[str, Type[Document]] = {
            ".txt": TxtProcessor,
            ".pdf": PdfProcessor,
```

```
            ".docx": DocxProcessor
        }

    def create_document(self, file_path: str) -> Document:
        ext = os.path.splitext(file_path)[-1].lower()
        if ext not in self.processors:
            raise ValueError(f"Unsupported file extension: {ext}")
        return self.processors[ext](file_path)

    def get_supported_formats(self) -> List[str]:
        return list(self.processors.keys())
```

Листинг 3: Класс DocumentFactory

3.3 Устройство настроек профиля

Профиль в данной системе представляет собой набор параметров, определяющих поведение системы при обнаружении и анонимизации персональных данных. Для хранения профилей был выбран JSON формат, который позволяет легко описывать различные аспекты настройки обработки данных. Пример простой настройки можно увидеть в листинге 4:

```
{
  "profile_id": "simple_profile",
  "use_regex": true,
  "use_dictionary": true,
  "use_language_model": true,
  "enabled_entity_types": ["PER", "PHONE"],
  "replacement_rules": {
    "PER":   { "type": "template", "template": "[PERSON]" },
    "PHONE": { "type": "remove" }
  },
  "custom_entity_prompts": {
    "PER":   "names and surnames, for example Ivan Ivanov",
    "PHONE": "phone numbers in any format, for example +7 123
   456-78-90"
  },
  "dictionary_paths": {
    "surnames": {
      "path": "dictionaries/surnames.txt",
      "entity_type": "PER",
      "enabled": true
    }
  },
  "llm_settings": {
    "model_path": "/path/to/local/Vikhr-Gemma-2B-instruct",
```

```
    "device": "cuda:0",
    "max_input_tokens": 512,
    "chunk_overlap_tokens": 50,
    "max_new_tokens": 768,
    "temperature": 0.5
  }
}
```

Листинг 4: Пример настроек

3.3.1 Описание параметров

1. **profile_id**: Уникальный идентификатор профиля, используется для выбора конкретных настроек.

2. **use_regex**: Флаг, указывает, нужно ли использовать регулярное выражение для обнаружения сущностей.

3. **use_dictionary**: Флаг, указывает, нужно ли использовать словарь для обнаружения сущностей.

4. **use_language_model**: Флаг, указывает, нужно ли использовать языковую модель для обнаружения сущностей.

5. **enabled_entity_types**: Список типов сущностей, которые поддерживаются в данном профиле (в данном примере например, “PER” - персона, “LOC” - местоположение).

6. **replacement_rules**: Правила замены для каждого типа сущности:

   - **type**: Тип замены (“template” для замены на шаблон, “remove” для удаления сущности, можно оставить пустым, тогда будет всё равно remove).
   - **template**: Шаблон замены, используется при типе “template”.

7. **custom_entity_prompts**: Настраиваемые подсказки (в каком-то роде определения) для типов сущности, для которых администратор хочет, чтобы языковая модель занималась поиском их.

8. **dictionary_paths**: Пути к словарям:

   - **path**: Путь к файлу словаря.
   - **entity_type**: Тип сущности, определенной в словаре.
   - **enabled**: Флаг, указывает, включен ли словарь для данного профиля.

9. **llm_settings**: Настройки языковой модели: Все эти настройки должны зависеть от возможности системы, поэтому каждой администратор должен выбрать подходящие настройки один раз.

- **model_path**: Путь до весов модели.
- **device**: Устройство для инференса (`"cpu"` или `"cuda:0"`).
- **max_input_tokens**: Максимальное количество токенов во входных данных.
- **chunk_overlap_tokens**: Количество перекрывающихся токенов между частями текста.
- **max_new_tokens**: Максимальное количество новых токенов для генерации текста.
- **temperature**: Параметр температуры для генерации случайных значений в языковой модели.

Эти параметры позволяют настроить систему анонимизации под конкретные требования и подзадачи анонимизации данных. Профелей может быть сколько угодно, в случае с серверной частью, они должны располагаться по пути api/config/profiles.json.

3.4 Применение языковой модели для контекстного анализа

При обработке текстовых данных система использует языковую модель для контекстного анализа и обнаружения персональных данных. Затем после анализа выхода языковой модели срабатывают дополнительные методы для повышения надёжности распознавания. Основной поток работы состоит из следующих этапов:

Формирование промпта и генерация разметки:
Для каждого текстового чанка (фрагмента длиной `max_input_tokens` с перекрытием `chunk_overlap_tokens`) строится промпт по примерно такому шаблону:

- Описание, его начало: Ты — алгоритм для анонимизации текста...
- Список типов тегов и их пользовательских описаний (`custom_entity_prompts`).
- Инструкция вернуть *весь* исходный текст, но с обёрткой найденных сущностей в теги.

Модель (Vikhr-Gemma-2B-instruct) при обработке фрагмента возвращает тот же текст, но с обёрнутыми в теги сущностями, например: <PER>Иван Иванович</PER> прибыл в <LOC>Москву</LOC> на встречу. Далее система простым регулярным выражением находит все вхождения вида <TYPE>...</TYPE> и по ним извлекает и сам текст сущности, и её тип. Начиная с этих позиций можно выполнить замену или удаление персональных данных, сохраняя исходную форму слов.

Извлечение сущностей из тегов. Для каждого обнаруженного тега система пытается найти точное вхождение разметки в исходном тексте. Если совпадение найдено, создаётся объект `Entity(text, type, start, end)`.

Извлечение через морфологию (spaCy). Если точное совпадение не найдено, но модель вернула некоторый кусок текста `ent_text` (модели часто любят использовать

начальную форму слова, например зачастую модель может найти сущность Москва, хотя в тексте могло быть в Москве или любой другой падеж), то прибегаем к библиотеке `spaCy` (модель `ru_core_news_sm`):

- Лемматизируется `ent_text` и весь документ.
- Выполняется поиск по совпадению лемм: любая лемма в документе, равная лемме `ent_text`, считается соответствующей сущностью (с сохранением оригинальной формы и позиции).

Извлечение через нечёткий поиск (RapidFuzz). Для текста, который не попал в два предыдущих шага, применяется библиотека RapidFuzz. Для каждого `ent_text` пробегаемся скользящим окном по исходному тексту длиной $|ent\_text| \pm N$. Затем вычисляется коэффициент совпадения Левенштейна. Если отношение совпадения больше значения (например 80%), то также фиксируем соответствующее вхождение. Это сделано для ситуаций, когда при написании слова в оригинале была допущена ошибка. Например вместо имени Анна написано Ан на, так как сотрудник случайно нажал на пробел. Языковая модель конечно же поймёт, что это имя, но запишет уже исправленную версию, что нам не совсем подходит. Нечёткий поиск решает эту проблему.
Таким образом, система сначала полагается на самую точную разметку LLM, затем пытается восстановить форму слов через морфологический анализ, и лишь в крайних случаях использует нечёткий поиск для распознавания и анонимизации сущностей. Этот подход обеспечивает высокую точность и полноту результатов.

3.5 Реализация веб-интерфейса

Веб-интерфейс системы FreeVigilanceReduction разработан для обеспечения простого и удобного взаимодействия оператора с функционалом анонимизации с Backend (Python + FastAPI[7]). FastAPI для реализации серверной части, обеспечивающей взаимодействие с клиентской частью через REST API. Клиентская часть построена с использованием стандартных веб-технологий, включая HTML для структуры страницы, CSS для стилизации элементов интерфейса и JavaScript для интерактивности и обработки событий. На рисунке 5 представлен главный экран системы.

Оператор может загружать текстовые документы различных форматов (TXT, DOCX, PDF) для их последующей обработки. При этом интерфейс предоставляет возможность выбора профиля анонимизации из заранее настроенных. Перед скачиванием итогового документа оператор может просмотреть его содержимое и внести корректировки в результаты анонимизации. Это позволяет оперативно проверить и изменить результаты, если это необходимо. При этом есть возможность переключаться между разными файлами, это реализовано через вкладки (их названия можно спокойно редактировать для удобства). После обработки документа система также отображает мини отчёт с списком обанруженных сущностей и их тегов. Это позволяет оператору быстро оценить результаты и убедиться в корректности анонимизации. После завершения обработки

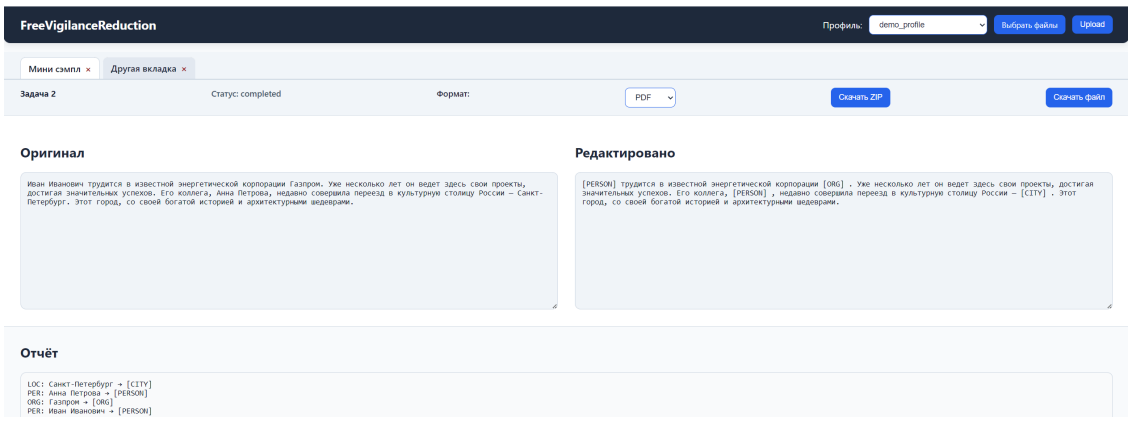

Рис. 5: Главный экран системы с функцией загрузки документов и предпросмотром результатов анонимизации

оператор может скачать анонимизированный документ в выбранном формате (TXT, DOCX, PDF), готовый для дальнейшего использования или передачи заказчику.

3.6 Разработка API

API системы предоставляет набор эндпоинтов для управления процессом анонимизации данных, может быть встроена в другие проекты.

`GET /profiles`

Возвращает список доступных профилей анонимизации с их идентификаторами, названиями и описаниями. Этот эндпоинт позволяет клиентским приложениям предоставлять пользователю выбор подходящего профиля в зависимости от типа задачи.

`POST /upload`

Основной эндпоинт для загрузки документов на обработку. Принимает файлы с указанием идентификатора профиля анонимизации. В ответ возвращается уникальный идентификатор задачи, который используется для отслеживания статуса обработки.

`GET /status/{taskId}`

Позволяет получить информацию о текущем статусе задачи обработки по её идентификатору.

`GET /results/{taskId}`

Получить список обработанных файлов и их отчётов по задаче.

`GET /download/{taskId}`

Сформировать zip-архив с результатами задачи и вернуть его.

3.7 Планирование и обработка результатов эксперимента

Для оценки качества анонимизации была проведена серия тестов на синтетических данных с использованием GPT-4o. Было сгенерировано 20 текстов объёмом от 1 до 3 страниц формата A4, имитирующих реальные документы (деловую переписку, отчёты, медицинские заключения). Все тексты были созданы с помощью промпт-инжиниринга, что позволило задавать примерно количество итоговых сущностей (чтобы у всех текстов их было примерно одинаково) и типы персональных данных. Применение синтетических данных было обусловлено невозможностью демонстрировать респондентам чужие реальные документы из-за ограничений конфиденциальности и этических норм.

Каждому из 9 респондентов случайным образом выдавались по 5 текстов для ручной анонимизации. Их результаты сравнивались с автоматической обработкой тем же набором из 20 текстов. Сравнение по времени не проводилось, поскольку скорость работы программы сильно зависит от аппаратных настроек и параметров LLM, а настройка каждого текста выполнялась так, чтобы среднее время машинного алгоритма было сопоставимо со временем, затраченным людьми для справедливости сравнения. Поскольку от запуска к запуску языковая модель может иначе реагировать на текст (за это отвечает параметр температуры, чем ниже, тем более уверенные и стабильные результаты получаются. Поэтому для данной задачи я бы рекомендовал небольшие значения порядка 0.1-0.2), то на каждом тексте алгоритм запускался 5 раз, затем все результаты усреднялись.

Для сравнения были выбраны метрики precision и recall. Precision в данном случае измеряет долю корректных отметок (сколько из помеченных фрагментов действительно являются персональными данными), а recall показывает полноту обнаружения (какой процент всех существующих в тексте персональных данных нашёлся).

Таблица 2: Precision и Recall для респондентов и автоматической модели

Участник	**Precision**	**Recall**
Респондент 1	98.3	95.6
Респондент 2	100.0	96.3
Респондент 3	95.0	97.7
Респондент 4	100.0	97.1
Респондент 5	100.0	95.8
Респондент 6	100.0	100.0
Респондент 7	98.3	96.0
Респондент 8	100.0	97.2
Респондент 9	100.0	95.1
Модель	**90.4**	**98.7**

В таблице 2 представлены как индивидуальные метрики респондентов, так и агрегированные оценки модели. Необходимо отметить, что основной целью данного эксперимента являлась проверка работоспособности программного решения, а не определении точности классификации.

Следует учитывать, что в рамках данной задачи эффективность метрик существен-

но зависит от множества факторов: качества настроек системы, лингвистической сложности анализируемого текста и других параметров. Примечательно, что разработанный метод демонстрирует тенденцию к избыточному выделению информации по сравнению с людьми (людям менее свойственно отмечать что-то лишнее), но это соответствует приоритетной задаче — максимизации покрытия персональных данных. В контексте защиты персональной информации критически важно минимизировать риск пропуска конфиденциальных данных, даже если это приводит к некоторому количеству ложноположительных результатов. Такой подход оправдан, поскольку: Невыявленные персональные данные представляют значительно больший риск, чем избыточно выделенная информация. Избыточные метки могут быть эффективно устранены на последующем этапе контрольной валидации с участием человека. При этом с Reccall (полнотой) всё не хуже людей.

3.8 Выводы по главе

В данной главе были представлена детальная программная реализация FreeVigilanceReduction[9] для автоматической анонимизации персональных данных в текстовых документах. Разработана Python-библиотека, реализующая подход к обнаружению персональных данных с помощью языковой модели Vikhr-Gemma-2B-instruct. Разработанный REST API обеспечивает возможность интеграции системы в существующие корпоративные решения. Реализован веб-интерфейс для удобного взаимодействия с системой.

Эта система может быть успешно применена в различных сферах, включая защиту данных в медицинских отчетах, финансовых документах и других чувствительных информационных ресурсах, обеспечивая высокий уровень безопасности и конфиденциальности данных.

4 ЗАКЛЮЧЕНИЕ

В данной работе был разработан сервис для защиты персональных данных в текстовых документах на русском языке. Разработанная система делает упор на обнаружение персональных данных с помощью языковой модели. Реализован весь заявленный функционал. Цель была достигнута, а все задачи выполнены. Итоговое решение можно использовать как автономный сервис для организаций, работающих с конфиденциальными документами, так и в качестве интегрируемой библиотеки или API для существующих корпоративных систем.

В качестве направления дальнейшего развития предполагается расширение поддержки форматов документов, добавление моделей для других языков, а также улучшение точности обнаружения сущностей за счет применения более специализированных моделей машинного обучения. Дополнительно рассматривается возможность встроенного дообучения модели (пользователи смогут оценивать качество работы алгоритма, что можно будет использовать в будущем, как данные для модели).

Список литературы

1. Кодекс Российской Федерации об административных правонарушениях от 30.12.2001 № 195-ФЗ (ред. от 03.02.2025): статья 13.11 // КонсультантПлюс.
2. Уголовный кодекс Российской Федерации от 13.06.1996 № 63-ФЗ (ред. от 20.03.2025): статья 137 // КонсультантПлюс.
3. Федеральный закон «О персональных данных» от 27.07.2006 № 152-ФЗ (последняя редакция) // КонсультантПлюс.
4. Anonympy [Электронный ресурс] // GitHub. Режим доступа: https://github.com/ArtLabss/open-data-anonymizer свободный. (дата обращения: 25.03.2025).
5. Datanymizer [Электронный ресурс] // GitHub. Режим доступа: https://github.com/datanymizer/datanymizer свободный. (дата обращения: 25.03.2025).
6. Devlin, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding // arXiv:1810.04805, 2018.
7. FastAPI [Электронный ресурс] // FastAPI. Режим доступа: https://fastapi.tiangolo.com/ свободный. (дата обращения: 21.03.2025).
8. Gemma Team et al. Gemma: Open Models Based on Gemini Research and Technology // arXiv:2403.08295, 2024. URL: https://arxiv.org/abs/2403.08295 (дата обращения: 27.03.2025).
9. GitHub-репозиторий FreeVigilanceReduction [Электронный ресурс] // GitHub. Режим доступа: https://github.com/FreeVigilance/PD-Reduction свободный. (дата обращения: 25.04.2025).
10. Honnibal, M. & Montani, I. spaCy: Industrial-Strength Natural Language Processing in Python [Электронный ресурс] // Explosion AI GmbH. Режим доступа: https://spacy.io свободный. (дата обращения: 26.04.2025).
11. Jiang, A. Q. et al. Mixtral of Experts // arXiv:2401.04088, 2024. URL: https://arxiv.org/abs/2401.04088 (дата обращения: 28.03.2025).
12. Lingvanex Document Anonymization [Электронный ресурс] // Lingvanex. Режим доступа: https://lingvanex.com/ru/technologies/data-anonymization-tool/ свободный. (дата обращения: 25.03.2025).
13. Microsoft Presidio [Электронный ресурс] // GitHub. Режим доступа: https://github.com/microsoft/presidio свободный. (дата обращения: 25.03.2025).
14. Radford, A. et al. Language Models are Unsupervised Multitask Learners // OpenAI Blog, 2019. URL: https://openai.com/research/better-language-models (дата обращения: 28.03.2025).

15. Sber DeepDocs [Электронный ресурс] // SberCloud. Режим доступа: https://deepdocs.sberlegal.ru/ свободный. (дата обращения: 25.03.2025).

16. Tjong Kim Sang, E. F. & De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition // Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003. 2003. P. 142–147. URL: https://aclanthology.org/W03-0419 (дата обращения: 28.03.2025).

17. Vikhr-Gemma-2B-instruct [Электронный ресурс] // Hugging Face. Режим доступа: https://huggingface.co/Vikhrmodels/Vikhr-Gemma-2B-instruct свободный. (дата обращения: 20.03.2025).

ПРИЛОЖЕНИЕ А

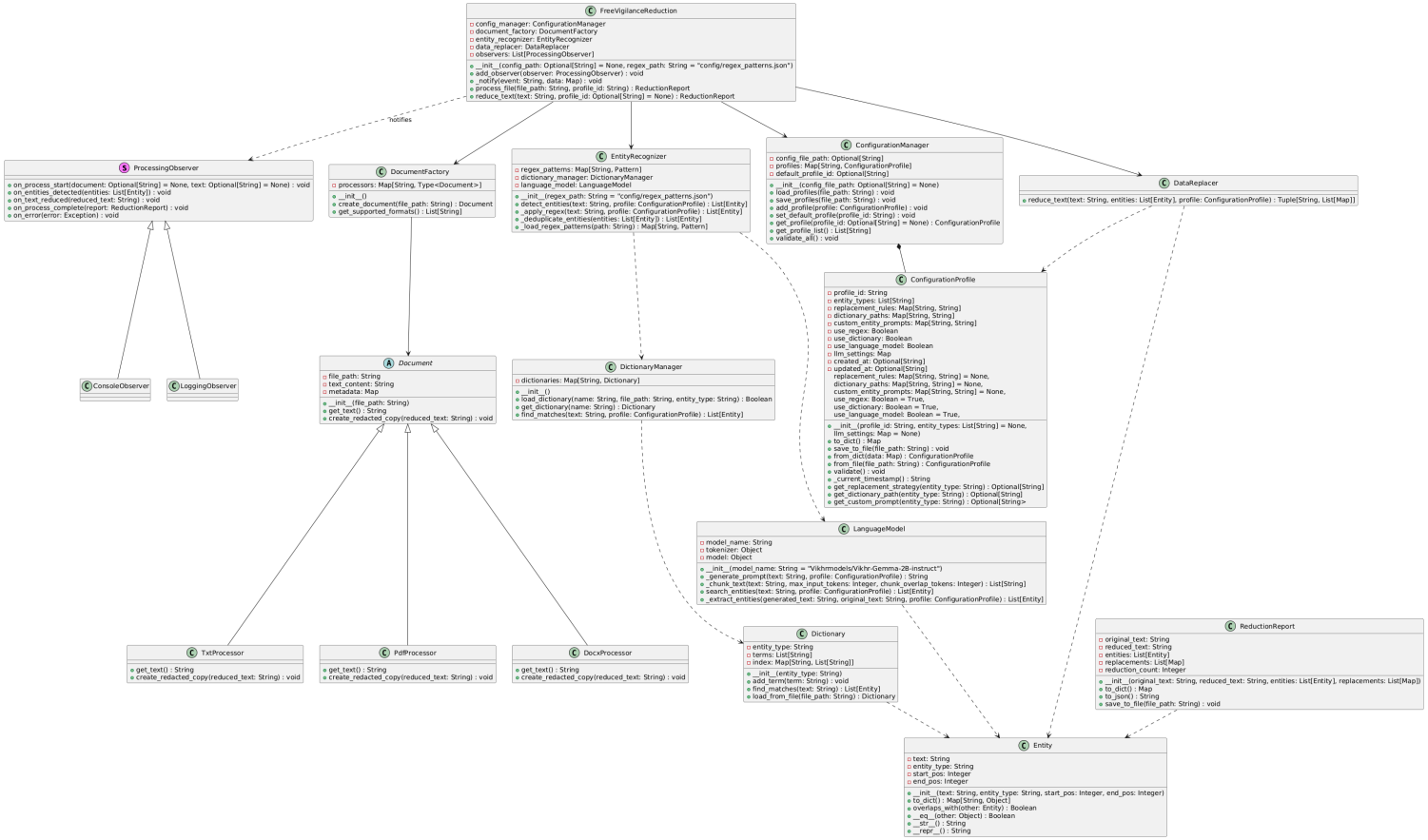

Рис. 6: Диаграмма классов Python-библиотеки анонимизации