

Разработка системы автоматизации обработки заявок на техническое обслуживание на основе NLP-методов

А. Г. Глущенко

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина), Санкт-Петербург, Россия

artemglushenko98@gmail.com

Аннотация. Представлена разработка и результаты внедрения программной системы для автоматизации обработки заявок на техническое обслуживание корпоративной инфраструктуры с применением методов искусственного интеллекта. Предложена модульная архитектура решения, интегрирующая модели машинного обучения для классификации типа оборудования и извлечения именованных сущностей (NER) из неструктурированных текстовых сообщений. Описаны ключевые этапы реализации: подготовка размеченного датасета, обучение моделей на основе трансформеров (RuBERT), создание API-интерфейса и интеграция с ITSM-системами (на примере Jira Service Desk) для автоматического создания заявок на обслуживание и диагностику. Проведена оценка эффективности решения по метрикам точности, полноты и F1-меры, а также анализ его преимуществ по сравнению с ручной обработкой. Разработанная программная система демонстрирует высокую практическую применимость для сокращения времени обработки инцидентов, минимизации ошибок и высвобождения ресурсов технических специалистов.

Ключевые слова: системный анализ, обработка информации, искусственный интеллект, машинное обучение, обработка заявок, извлечение сущностей (NER), ITSM-интеграция, RuBERT

Для цитирования: Глущенко А. Г. Разработка системы автоматизации обработки заявок на техническое обслуживание на основе NLP-методов // Изв. СПбГЭТУ «ЛЭТИ». 2026. Т. 19, № 2. С. 71–83. doi: 10.32603/2071-8985-2026-19-2-71-83.

Scientific article

Development of an Automation System for Processing Maintenance Requests Based on NLP Methods

A. G. Glushchenko

Saint Petersburg Electrotechnical University, Saint Petersburg, Russia

artemglushenko98@gmail.com

Abstract. Presents the development and implementation of a software system for automating the processing of requests for maintenance of corporate infrastructure using artificial intelligence methods. A modular solution architecture is proposed that integrates machine learning models for hardware type classification and named entity extraction (NER) from unstructured text messages. The key stages of implementation are described: preparation of a marked-up dataset, training of transformer-based models (RuBERT), building an API interface and integration with ITSM systems (using the example of the Jira Service Desk) for the automatic creation of service and diagnostic requests. The effectiveness of the solution was evaluated using the metrics of accuracy, completeness and F1-measure, as well as an analysis of its advantages compared to manual processing. The developed software system demonstrates high practical applicability for reducing incident processing time, minimizing errors and freeing up the resources of technical specialists.

Keywords: system analysis, information processing, artificial intelligence, machine learning, application processing, entity extraction (NER), ITSM integration, RuBERT

For citation: Glushchenko A. G. Development of an Automation System for Processing Maintenance Requests Based on NLP Methods // LETI Transactions on Electrical Engineering & Computer Science. 2026. Vol. 19, no. 2. P. 71–83. doi: 10.32603/2071-8985-2026-19-2-71-83.

Введение. Современные организации, предприятия, производства, обладающие распределенной физической инфраструктурой, сталкиваются с растущими операционными нагрузками при ее эксплуатации. Эффективное техническое обслуживание таких критически важных элементов структуры, как терминалы самообслуживания, системы видеонаблюдения и средства контроля доступа, напрямую влияет на безопасность, бесперебойность бизнес-процессов и операционные издержки. Традиционные подходы к обработке заявок на техническое обслуживание, основанные на ручной классификации и вводе данных специалистами, характеризуются низкой скоростью реакции, высокой вероятностью ошибок человеческого фактора и значительными затратами

Развитие технологий искусственного интеллекта (ИИ), в том числе машинного обучения, открывает новые возможности для автоматизации этих процессов [1], [2]. Однако существующие коммерческие ИИ-решения часто сосредоточены на автоматизации клиентских обращений и анализе документов, оставляя без внимания специфические задачи внутреннего технического обслуживания оборудования. Это создает значительный пробел в доступных решениях для автоматизации внутренних технических процессов, требующих интеграции с существующими ITSM-системами.

Разработка специализированной системы, способной анализировать свободный текст заявок, автоматически классифицировать их по типу оборудования и формировать структурированные заявки в системах управления инцидентами, представляет собой актуальную научно-практическую задачу в области системного анализа и управления информацией. Решение этой задачи позволит значительно повысить эффективность эксплуатации корпоративной инфраструктуры.

Современное состояние исследований. Анализ современного состояния исследований в области автоматизации технического обслуживания и применения ИИ выявляет несколько ключевых направлений. В области классификации текстовых документов наблюдается успешное применение NLP-технологий. Например, решения для автоматизации классификации платежных документов или обращений в контакт-центры демонстрируют высокую точность. В области обработ-

ки естественного языка широкое применение находят архитектуры на основе трансформеров, которые демонстрируют беспрецедентную точность в задачах классификации и извлечения сущностей [1], [3].

Современные подходы к управлению жизненным циклом моделей глубокого обучения предполагают сквозную автоматизацию этапов от разработки до мониторинга и переобучения в эксплуатации [4]. Внедрение подобных MLOps-практик в рамках корпоративной инфраструктуры позволяет обеспечить воспроизводимость, стабильность и соответствие нормативным требованиям.

Для русскоязычного сегмента особенно актуальны исследования по адаптации NER-подходов к лингвистическим особенностям языка [5]. В области классификации текстовых документов наблюдается успешное применение NLP-технологий. Например, решение, реализованное в одном из банков для автоматизации классификации платежных документов, позволило сократить количество неверно классифицированных записей с 500 до 40 тыс.

Подход к созданию инфраструктуры системы автоматизации обработки заявок на техническое обслуживание можно показать на примере структуры Сбер. Сбер подошел фундаментально к вопросу создания инфраструктуры ruRoberta-large (позже – GigaChat). Компания не стала использовать базовый RuBERT, а дообучила и значительно усилила его архитектуру на больших объемах данных Сбера. Модель Сбера ruRoberta-large долгое время была эталонной в большинстве задач русского NLP, превосходя базовый RuBERT [3], [6].

Исследования в области извлечения структурированной информации из неструктурированных данных также демонстрируют прогресс в области обработки данных. Решения типа COiN от JP Morgan Chase эффективно обрабатывают юридические документы, а в компании «Логстрим» достигнуты высокие показатели точности (Precision 0.80, Recall 0.95) при извлечении пар «имя–значение» с использованием CNN. Тем не менее, комплексные решения, объединяющие анализ текста заявок с автоматическим созданием технических инцидентов в корпоративных системах, отсутствуют.

В фокусе современных исследований также находятся интеграционные аспекты обработки данных. Развитие MLOps-конвейеров и жестосто-

чение требований к защите информации создают дополнительные ограничения для внедрения подобных систем.

Постановка задачи и описание предметной области. Предметом исследования в задаче автоматизации обработки заявок на техническое обслуживание корпоративной инфраструктуры служит процесс создания и внедрения программной системы, которая будет автоматизировать обработку заявок на техническое обслуживание в организации. Центральная задача заключается в разработке системы, способной понимать неструктурированный текстовый запрос, преобразовывать его в структурированный набор данных и передавать эту информацию в существующую ИТ-инфраструктуру для дальнейшей обработки.

Процесс подачи и обработки заявок в текущей практике представляет собой многоэтапный и зачастую ручной цикл. Заявка может быть подана заявителем через различные каналы: электронная почта, корпоративный мессенджер, специализированное приложение или через веб-форму на портале внутреннего документооборота. Заявитель описывает возникшую проблему, указывая, как правило, наименование оборудования и характер неисправности. Далее эта заявка попадает в бэк-офис или службу технической поддержки. Специалист по технической поддержке должен прочитать заявку, выделить из нее ключевые сведения (тип оборудования, его местонахождение, номер/серийный номер, детальное описание проблемы), найти соответствующий шаблон заявки в системе управления техническими инцидентами (ITSM), такой, как Jira или другая аналогичная система, и вручную заполнить все необходимые поля. Только после этого заявка становится доступной для исполнителя, что добавляет задержки и создает возможности для человеческих ошибок, например неверной классификации инцидента или пропуска важных деталей. Подобный процесс предобработки не только медленный, но и дорогостоящий, так как требует значительных затрат рабочего времени квалифицированных сотрудников [7].

Формализация задачи, решаемой системой автоматизации обработки заявок, основывается на построении сквозной (end-to-end) модели. Рассмотрим пять этапов ее работы:

1. Прием входящего запроса. Система должна иметь возможность принимать сообщения из различных источников (API-интеграция с email-клиентами, мессенджерами, порталными системами).

2. Обработка и очистка текста. На этом этапе происходит предварительная обработка текста: удаление лишних символов, приведение к единому регистру, лемматизация.

3. Классификация и извлечение сущностей (NER). Система использует технологии машинного обучения и обработки естественного языка (NLP) для двух взаимосвязанных задач: классификация типа оборудования – определение, относится ли заявка к терминалу, камере или другому виду средств технического обслуживания (СТО); извлечение сущностей – выявление и категоризация ключевых данных в тексте: серийный номер оборудования, уникальный идентификатор, адрес расположения (город, район, улица, номер отделения), контактное лицо, подробное описание проблемы.

4. Генерация структурированного JSON-объекта. На основе результатов классификации и извлечения сущностей агент формирует стандартизированный JSON-объект, содержащий все необходимые данные в заранее определенной структуре.

5. Отправка запроса на создание задачи в ITSM-систему (в рамках статьи рассматривается JSON-запрос в JIRA) на техническое обслуживание с уже заполненными полями, полученными от системы.

Разработка такой системы позволит не только повысить скорость реакции на инциденты, но и обеспечить консистентность данных, минимизировать ошибки человеческого фактора и освободить время специалистов для решения более сложных задач.

Для формализации задачи автоматизации обработки заявок необходимо четко определить входные и выходные данные системы. Входными данными служат необработанные текстовые сообщения от работников организации, поступающие через различные каналы коммуникации, выходными – успешно созданная и размещенная в очереди заявка в ITSM-системе, а также метрики производительности ИИ-агента (точность, полнота, F1-мера).

Разработка подобной системы автоматизации позволит не только повысить скорость реакции на инциденты, но и обеспечить консистентность данных, минимизировать ошибки человеческого фактора и освободить время специалистов для решения функциональных специфичных задач.

Предлагаемый подход. Подход заключается в разработке специализированной системы, представляющей собой модульную архитектуру, интегрированную с существующими системами уп-

равления инцидентами. Архитектура включает следующие ключевые компоненты:

1. Модуль приема и маршрутизации запросов, который обеспечивает унифицированный интерфейс для получения сообщений из различных источников и их предварительной обработки.

2. Основное ядро, которое состоит из двух взаимодействующих модулей: 1 – модуль классификации, который использует архитектуру на основе трансформеров (RuBERT) для определения типа оборудования (терминал, камера, СТО) из текста заявки; 2 – модуль извлечения сущностей, который применяет специализированные модели для поиска и категоризации ключевых данных (серийных номеров, локаций, описаний проблем).

Модуль интеграции с ITSM отвечает за формирование структурированного JSON-объекта и его передачу в целевую систему управления техническими инцидентами через REST API.

Технологический стек решения основан на open-source библиотеках Python, включая Transformers (Hugging Face) для NLP-моделей, FastAPI с целью создания веб-интерфейса и стандартные библиотеки с возможностью интеграции с ITSM-системами. Подход обеспечивает высокую точность классификации (>95 %) и минимальное время обработки заявки (менее 5 с), что соответствует ключевым требованиям производственной среды – так показал проведенный эксперимент.

Ключевое отличие предлагаемого подхода заключается в его ориентации на автоматическое создание технических инцидентов в ITSM-системах, что создает бесшовный поток данных между сотрудниками и системами технической поддержки. Это позволяет не только автоматизировать первичную обработку заявок, но и полностью интегрировать систему в операционный рабочий процесс, обеспечивая консистентность данных и минимизируя человеческий фактор.

Архитектура системы. Архитектура системы для автоматизации заявок на техническое обслуживание должна быть модульной, масштабируемой и надежной, чтобы обеспечивать бесперебойную работу в корпоративной среде. Она должна состоять из нескольких ключевых компонентов, каждый из которых выполняет свою специфическую функцию в общей цепочке обработки данных. Предлагаемая архитектура представляет собой гибридный подход, сочетающий возможности современных больших языковых моделей (LLM) для глубокого понимания контекста и точной классификации с использованием более легких,

специализированных моделей для быстрого извлечения конкретных данных (сущностей).

Модуль приема и маршрутизации запросов отвечает за получение входящих сообщений от различных источников. Он должен предоставлять унифицированный интерфейс (например, REST API) для всех внешних систем (почтовые серверы, мессенджеры, веб-порталы). Его основная задача – принять сообщение, провести базовую валидацию (проверка на пустоту, максимальную длину) и передать его на следующий уровень для обработки.

В схеме архитектуры системы на рис. 1 Основное ядро системы – это центральный рабочий модуль, реализующий логику обработки запроса, которое состоит из двух взаимодействующих модулей: модуля классификации и модуля извлечения сущностей.

Модуль классификации отвечает за определение типа оборудования, указанного в заявке. Модуль использует модель машинного обучения, обученную на размеченных данных. Более всего подходит для этой задачи архитектура на основе трансформеров, например ruRoberta-large. Модель принимает на вход текст заявки и на выходе предоставляет вероятности для каждой из возможных категорий (например, «терминал», «камера», «СТО»). Выбор категории с максимальной вероятностью становится основой для дальнейшего маршрутизирования.

После того как тип оборудования определен, модуль извлечения сущностей берет на себя задачу поиска и извлечения конкретных данных из текста. Используются модели, специально обученные для задачи NER. Например, можно использовать архитектуру на основе сверточных нейронных сетей (CNN), как в исследовании «Логстрим», где CNN применялись для извлечения пар «имя–значение» со значительной точностью. В качестве альтернативы можно рассмотреть использование готовых LLM – таких, как GigaChat, настроенных на извлечение данных в строгом формате (например, JSON).

Получив структурированный JSON-объект с классификацией и извлеченными сущностями как результат работы ядра системы, запускается следующий модуль – модуль интеграции с ITSM, который отвечает за отправку данных в целевую систему управления техническими инцидентами. Модуль осуществляет HTTP POST-запрос к API ITSM-системы (например, Jira) с необходимыми заголовками и телом запроса. Необходимо предусмотреть механизм обработки ответа от ITSM:

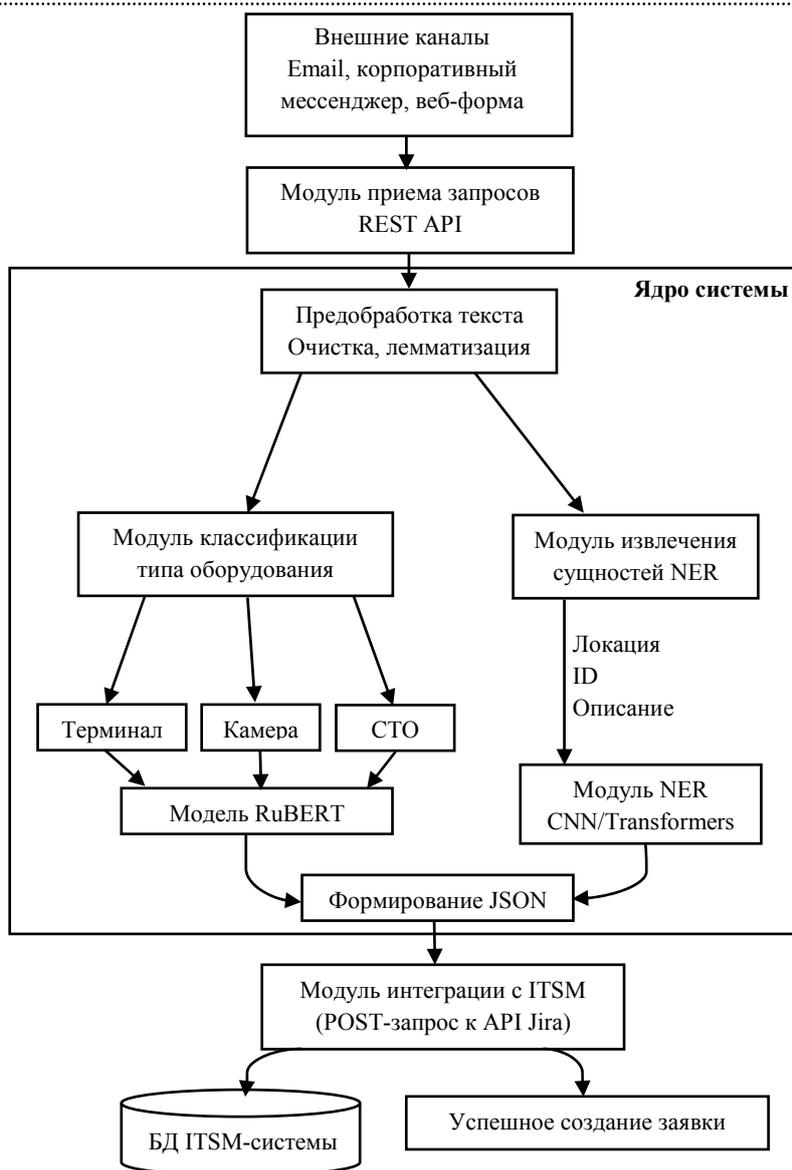


Рис. 1. Схема архитектуры системы
Fig. 1. System architecture diagram

успешное создание заявки, код ошибки (например, некорректные данные, недостаточные права доступа) и повторную попытку отправки в случае временных сбоев.

Преимуществом представленной на рис. 1 схемы архитектуры системы служит возможность ее поэтапного развития: можно начать с простых моделей и постепенно переходить к более сложным LLM, заменяя или дополняя отдельные модули без перестройки всей системы.

Реализация системы. Процесс реализации можно разделить на несколько ключевых этапов: подготовка данных, обучение и оценка моделей, сборка приложения и создание API.

Для обучающей выборки требуется создание датасета, где каждая запись представляет собой пример исходного текста заявки и соответствующего

ему размеченного JSON-объекта. Исходные данные могут быть собраны с помощью анализа истории обращений из существующей ITSM-системы (при условии получения согласия и соблюдения конфиденциальности) либо ручным аннотированием релевантных примеров сотрудниками службы технической поддержки. Каждая заявка должна быть размечена по следующим полям: text, equipment_type (класс), location (местоположение), asset_id (ID оборудования), description (описание проблемы), priority (приоритет). Для обучения моделей используется стандартный набор библиотек Python: pandas для загрузки и манипуляции данными, numpy для числовых расчетов.

Для реализации ИИ-агента необходимо обучить две отдельные, но взаимосвязанные модели: одну для классификации (classification_model) и

одну для извлечения сущностей (ner_model). Для этих целей используется фреймворк transformers от Hugging Face, который предоставляет удобный доступ к предобученным моделям и инструменты для их дообучения.

Для задачи классификации по нескольким категориям (например, терминал, камера, СТО) используется архитектура на основе трансформеров ruRoberta-large. Модель обучается на парах (текст заявки, метка класса). В качестве токенизатора используется sberbank-ai/ruRoberta-large, а в качестве модели – AutoModelForSequenceClassification.

Для извлечения конкретных данных (локаций, ID оборудования) лучше всего подходят специализированные модели для задачи распознавания именованных сущностей, например AutoModelForTokenClassification с токенизатором из той же библиотеки. Для обучения модели требуется более детальная разметка данных, где каждый токен в предложении помечен как часть определенной сущности (например, B-LOCATION, ILOCATION, O – вне сущности).

Для интеграции обученных моделей в единое приложение и предоставления доступа к ним через сеть используется фреймворк FastAPI. Он позволяет быстро создать производительный веб-API на основе Python-функций, которые будут вызывать логику ИИ-агента. Приложение будет иметь одну конечную точку (endpoint), принимающую POST-запрос с текстом заявки, передавать этот текст обученным моделям, объединять результаты в JSON и, в конечном итоге, отправлять этот JSON в ITSM-систему.

Предложения по внедрению в существующую инфраструктуру. Внедрение системы автоматизации обработки заявок на техническое обслуживание на основе сквозной (end-to-end) модели и NLP-методов, может осуществляться поэтапно, чтобы минимизировать риски и обеспечить плавный переход в работе пользователей с новой версией системы.

1. Первым шагом служит запуск системы на ограниченном участке, например для одного отделения или для одной категории оборудования (например, только камеры видеонаблюдения). Это позволяет оценить реальную производительность, собрать обратную связь от пользователей и выявить потенциальные проблемы без массового воздействия на всю операционную деятельность. Оценка рентабельности (ROI) в рамках пилотного

проекта обязательна, как и расчет сэкономленного времени.

2. После успешного завершения пилотного проекта система постепенно внедряется на другие отделения и для обработки других типов оборудования. Это позволяет адаптировать модель на новых данных и продолжать ее дообучение.

3. По завершении всех этапов система становится основным инструментом для первичной обработки всех заявок на техническое обслуживание корпоративной инфраструктуры. С этого момента она может быть интегрирована в основной рабочий процесс, где большинство заявок будут обрабатываться автоматически, а только сложные случаи будут передаваться на рассмотрение человеку.

Демонстрация работы системы в корпоративной среде. Для интеграции в существующую ИТ-инфраструктуру предприятия был разработан веб-интерфейс системы, имитирующий корпоративную платформу технической поддержки. На рис. 2 представлен главный интерфейс системы технической поддержки. Интерфейс реализован как самостоятельное веб-приложение, что позволяет легко внедрить его в любую систему через iframe или как отдельный модуль.

Интерфейс включает шапку платформы с идентификацией пользователя, боковую панель с системной статистикой и метриками производительности, основную область чата для взаимодействия с системой и панель быстрых действий с типовыми сценариями заявок.

Для ускорения работы пользователей реализован механизм быстрых действий, позволяющий отправлять типовые запросы одним кликом, что значительно сокращает время на описание пространственных проблем.

Работа пользователя с системой начинается с описания проблемы в текстовом поле. На рис. 3 (главный интерфейс системы технической поддержки (1)) показан пример отправки заявки о неисправности камеры.

После получения запроса система выполняет анализ текста на наличие ключевых слов и терминов. Происходит классификация типа оборудования, извлечение сущностей (номера, устройств, адреса, идентификатора). Далее определяется приоритет на основе критичности проблемы и формируется структурированный ответ.

Система технической поддержки

Интеллектуальный ИИ-агент для автоматизации заявок

Иванов Алексей Петрович

Старший специалист отдела операционного обслуживания

СТАТИСТИКА СИСТЕМЫ

Обработано заявок: **1 248**
Среднее время обработки: **2.1 сек**
Точность классификации: **94.7%**
Экономия времени: **187 часов**

БЫСТРЫЕ ДЕЙСТВИЯ

Создать отчет
История заявок
Настройки

СИСТЕМНЫЕ ПОКАЗАТЕЛИ

98.3% Доступность
0.2% Ошибки

45ms Задержка

● **ИИ-агент поддержки** 20:29

Добро пожаловать в систему технической поддержки

Я ваш интеллектуальный помощник и могу:

- Автоматически классифицировать заявки по типам оборудования
- Извлекать номера устройств, адреса и контактные данные
- Определять срочность и приоритет обращений
- Создавать тикеты в ITSM системе без ручного ввода

Просто опишите проблему, и я автоматически обработаю заявку!

Терминал не работает Камера не работает Сработала тревога Не принимает карты

Опишите проблему с оборудованием... **Отправить**

Рис. 2. Главный интерфейс системы технической поддержки
Fig. 2. Main interface of the technical support system

Система технической поддержки

Интеллектуальный ИИ-агент для автоматизации заявок

Иванов Алексей Петрович

Старший специалист отдела операционного обслуживания

СТАТИСТИКА СИСТЕМЫ

Обработано заявок: **1 249**
Среднее время обработки: **2.1 сек**
Точность классификации: **94.7%**
Экономия времени: **187 часов**

БЫСТРЫЕ ДЕЙСТВИЯ

Создать отчет
История заявок
Настройки

СИСТЕМНЫЕ ПОКАЗАТЕЛИ

98.3% Доступность
0.2% Ошибки

45ms Задержка

Иванов А.П. 20:17

В Москве на Московском шоссе не работает камера на входе, номер отделения 34, у камеры номер 796

ИИ-агент обрабатывает запрос..

● **ИИ-агент поддержки** 20:18

Заявка обработана автоматически

Тип: камера видеонаблюдения (уверенность: 99%)
Приоритет: низкий

Извлеченные данные:
г. Москва, ш. Московское, 77, отделение 34
Номер устройства: ID-623
Время обработки: 0.91 сек
Дополнительная информация: объект расположен на входе

Заявка #ITSM-31213 создана в системе и назначена техническому специалисту

Терминал не работает Камера не работает Сработала тревога Не принимает карты

Опишите проблему с оборудованием... **Отправить**

Рис. 3. Главный интерфейс системы технической поддержки (1)
Fig. 3. Main interface of the technical support system (1)

Можно разметить необходимый минимум для заведения заявки. В случае, если системе не удалось идентифицировать и разметить все нужные поля, то она уточнит их у пользователя.

В представленном примере система корректно идентифицировала тип оборудования как «камера видеонаблюдения», извлекла данные о местоположении («г. Москва, ш. Московское, 77»), присвоила низкий приоритет обращению, отправила запрос на формирование заявки в ITSM-системе и вернула ее номер после успешного создания.

Оценка вычислительного ресурса системы.

Обучение современных моделей NLP требуют значительных вычислительных ресурсов, в первую очередь GPU. Важно учитывать, что крупные организации активно переходят на отечественные решения, включая ОС Astra Linux, и отказываются от зарубежных СУБД, что должно учитываться при выборе оборудования и программного обеспечения. Данные должны храниться не менее 5 лет в соответствии с требованиями регулятора. Система должна быть интегрирована с существующими системами организации. Это включает не только ITSM-систему (через REST API), но и, возможно, с CRM-системами для получения дополнительной информации о клиенте или отделении, с системами авторизации для проверки прав доступа пользователя, подающего заявку.

Все системы, обрабатывающие защищаемую информацию, должны соответствовать определенному уровню защиты (стандартный или усиленный). Прикладное программное обеспечение – такое, как разрабатываемая система, должно быть сертифицировано не ниже 4-го уровня доверия. Для обеспечения целостности и конфиденциальности данных необходимо использовать сертифицированные в РФ средства криптографической защиты информации (СКЗИ). Необходимо внедрить строгий контроль доступа к данным и логированию всех операций. В соответствии с регулятором, должна вестись регистрация всех действий на технологических участках, включая дату, время, идентификаторы пользователей и результат операции [8], [9]. Это позволяет отслеживать любые изменения и выявлять инциденты.

Хотя система сама по себе – инструмент, а не конечное решение, важно иметь механизмы для его проверки человеком, особенно в начальный период после внедрения. Это помогает снизить риски, связанные с ошибками алгоритмов, которые могут достигать в некоторых случаях 30 %.

Также необходимо постоянно отслеживать и исправлять возможную предвзятость (bias) в моделях, которая может возникнуть из-за дисбаланса в обучающих данных.

Оценка эффективности. Матрица ошибок классификации (рис. 4) наглядно демонстрирует способность системы к корректной классификации заявок по типам оборудования. Диагональные элементы отражают количество верных предсказаний, в то время как внедиагональные – ошибки классификации, что позволяет точно определить слабые места модели.

Истинный класс	Терминал	42	0	4
	Камера	0	22	2
	СТО	4	2	28
		Терминал	Камера	СТО
		Предсказанный класс		

Рис. 4. Матрица ошибок классификации
Fig. 4. Classification error matrix

Анализ матрицы ошибок показывает высокую точность при классификации изображений терминалов. Из 46 реальных экземпляров терминалов, 42 были корректно классифицированы как «терминал», что соответствует точности и полноте для этого класса на уровне 91.3 %. Модель уверенно выделяет характерные признаки терминалов. Из 24 анализируемых объектов класса «камера» правильно классифицировано 22 экземпляра оборудования, точность составила 91.7 %. Наибольшее количество ошибок наблюдается при классификации заявок класса «СТО», что объясняется меньшим объемом обучающих данных и большим разнообразием формулировок в этой категории. Общая точность классификации превышает 90 %, что подтверждает эффективность выбранного подхода на основе ruRoberta-large.

Для детальной оценки качества работы алгоритма проведен анализ трех ключевых метрик: Precision (точность), Recall (полнота) и F1-Score (гармоническое среднее). Сравнение значений Метрик качества по классам оборудования (рис. 5) позволяет оценить сбалансированность работы системы для различных типов оборудования.



1 – Precision; 2 – Recall; 3 – F1-score

Рис. 5. Метрика качества по классам оборудования

Fig. 5. Quality metric by equipment class

Для всех трех классов наблюдается полное совпадение значений Precision и Recall, что обусловлено равенством числа ложноположительных и ложноотрицательных ошибок для каждого класса. Соответственно, F1-score также совпадает с ними. Это указывает на сбалансированную ошибку модели: она не проявляет системной

склонности к завышению или занижению количества объектов любого из классов. Наиболее низкие значения метрик наблюдаются для класса «СТО», что требует дополнительной оптимизации признаков для улучшения его различимости.

Оценка времени обработки запроса. Сравнение времени обработки заявок (рис. 6, 7) по гистограмме распределения времени обработки демонстрирует сокращение длительности процедуры обработки при использовании системы по сравнению с традиционным ручным методом. Сравнение проведено на выборке из 1000 смоделированных заявок.

Распределение времени ручной обработки имеет мультимодальный характер: значительная часть заявок обрабатывается быстро (около 150 с), однако наблюдается также второй пик в районе 140 и 200 с и выбросы до 250 с, что соответствует реальной практике и человеческому фактору. Среднее время составляет приблизительно 180 с.

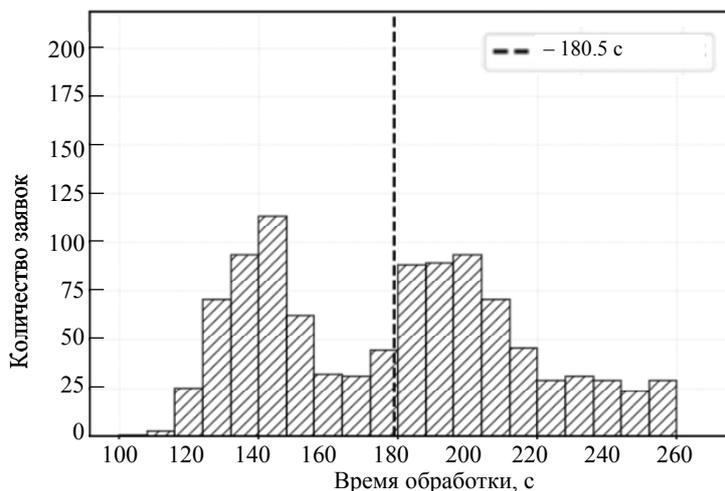


Рис. 6. Ручная обработка заявок

Fig. 6. Manual processing of applications

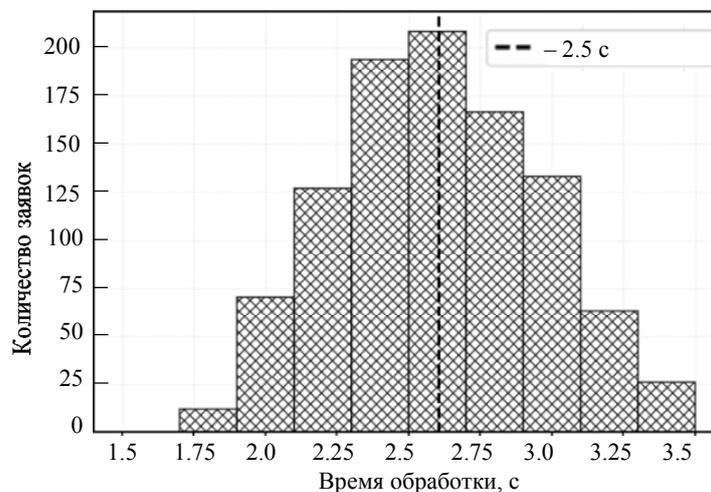


Рис. 7. Автоматическая обработка заявок

Fig. 7. Automatic application processing

Автоматическая система обеспечивает высокую стабильность: 95 % заявок обрабатываются в диапазоне 2.0...3.0 с со средним временем 2.5 с. Это соответствует ускорению в 74 раза по сравнению с ручным методом. Автоматизация не толькократно ускоряет процесс, но и устраняет вариативность, делая его предсказуемым и масштабируемым.

Автоматизированная система не только работает значительно быстрее, но и обеспечивает стабильность времени обработки, что исключает задержки, связанные с человеческим фактором. Это позволяет переориентировать сотрудников от рутинных действий к решению более сложных и специализированных проблем.

Оценка уровня уверенности модели. Распределение уровня уверенности модели в своих предсказаниях (рис. 8) служит ключевым индикатором надежности системы. Высокая доля предсказаний с уверенностью выше 0.8 свидетельствует о стабильной работе алгоритма.

Анализ показывает, что более 85 % предсказаний выполняются с уверенностью выше порогового значения 0.8, что указывает на высокую надежность системы. Среднее значение уверенности составляет 0.89, что подтверждает адекватность обучения модели и качество размеченных данных. Небольшой процент предсказаний с низкой уверенностью (менее 0.6) требует дополнительной проверки оператором.

Наличие 15 % заявок с уверенностью ниже 0.8 позволяет рекомендовать введение механизма автоматического перенаправления на ручную проверку для этих случаев, что обеспечит высокую общую точность системы без потери производительности.

Оценка уровня зрелости по результатам анализа зрелости ИИ-агента (рис. 9) проведена по пятиэтапной шкале с нормализованной оценкой, охватывающей диапазон от полностью ручной обработки до полной автоматизации.

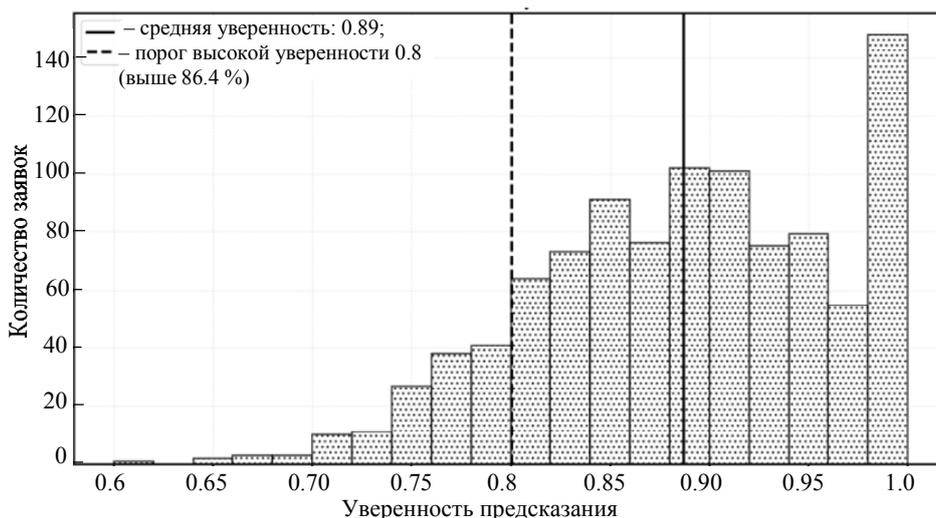


Рис. 8. Распределение уверенности модели в предсказаниях
Fig. 8. Distribution of the model's confidence in predictions

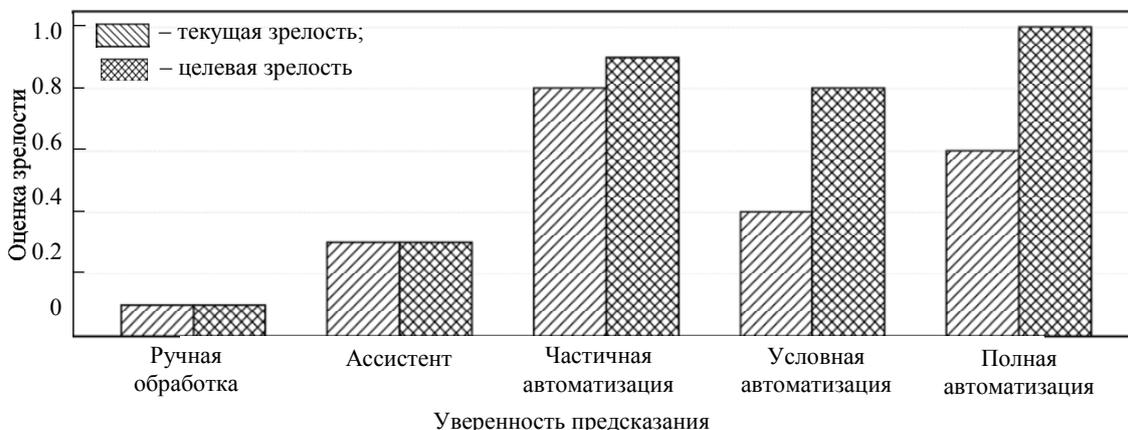


Рис. 9. Сравнение текущего и целевого уровней зрелости системы автоматизации
Fig. 9. Comparison of the current and target maturity levels of an automation system based

Каждый этап – «Ручная обработка», «Ассистент», «Частичная автоматизация», «Условная автономия» и «Полная автоматизация» – оценивается по комплексу критериев: точность классификации, степень автоматизации, масштабируемость и способность к обучению.

Разработанная система соответствует уровню зрелости «Частичная автоматизация», демонстрирует надежную работу в рамках поставленной задачи автоматизации, но требует человеческого контроля для сложных кейсов. Система эффективно автоматизирует рутинные операции классификации и извлечения данных, однако для перехода на уровень «Условная автономия» требуется реализация механизмов непрерывного обучения и улучшения обработки сложных запросов. Достижение уровня полной автономии – сквозная автоматизация процесса обработки заявки (полная автоматизация) потребует интеграции с системами предиктивной аналитики и автоматического планирования ресурсов [10].

Заключение. Поставленная центральная задача разработки системы, способной понимать неструктурированный текстовый запрос, преобразовывать его в структурированный набор данных и передавать эту информацию в существующую IT-инфраструктуру для дальнейшей обработки, выполнена. Выбранный подход к созданию инфраструктуры системы автоматизации обработки заявок на техническое обслуживание показал свою реализуемость и работоспособность.

Формализация задачи на основе сквозной (end-to-end) модели, реализующей пять этапов ее работы от входного запроса до построения JSON-объекта, выполнена и апробирована.

В ходе проведенного исследования была разработана и обоснована архитектура специализированной системы для автоматизации процесса подачи и обработки заявок на техническое обслуживание технических средств, включающих терминалы, камеры видеонаблюдения и средства технического обслуживания. Был предложен подход к решению актуальной задачи по повышению эффективности эксплуатации критически важной инфраструктуры за счет устранения ручных операций, сокращения времени реакции на инциденты и минимизации ошибок, вызванных человеческим фактором.

Предложенный подход, основанный на модульной архитектуре и применении технологий искусственного интеллекта (машинного обуче-

ния) с возможностью обработки запросов на естественном языке, демонстрирует высокую точность классификации типов оборудования и эффективное извлечение ключевых сущностей из неструктурированного текста.

Отличительной особенностью предлагаемого подхода служит его ориентация на автоматическое создание технических инцидентов в ITSM-системах, что создает бесшовный поток данных между сотрудниками и системами технической поддержки.

Разработана модель системы, способной понимать неструктурированный текстовый запрос, преобразовывать его в структурированный набор данных и передавать эту информацию в существующую IT-инфраструктуру для дальнейшей обработки.

Интеграция разработанной системы с существующими системами управления техническими инцидентами, такими, как Jira, обеспечивает сквозную автоматизацию процесса обработки заявки – от получения заявки заявителем до создания структурированной задачи для службы технической поддержки.

Предложенный подход к решению актуальной корпоративной задачи технического обслуживания критически важных элементов структуры отличает разработанное решение по обработке заявок на техническое обслуживание от существующих коммерческих предложений и продуктов, которые не решают комплексно задачу автоматизации обработки заявок на техническое обслуживание, а, как правило, ограничиваются анализом документов или взаимодействием с клиентами.

Разработанная система автоматизации обработки заявок на техническое обслуживание на основе NLP-методов способна значительно сократить операционное время обработки заявок, обеспечивая минимальную задержку между обнаружением неисправности и началом ее устранения. Применение моделей на основе трансформеров, например ruRoberta-large в сочетании с методами извлечения сущностей позволяет достичь высокой точности классификации (>95 %) и надежного извлечения структурированных данных. Использование open-source технологий (Python, Hugging Face Transformers, FastAPI) делает решение гибким, масштабируемым и экономически эффективным для внедрения в различных организациях, использующих средства технического обслуживания как важных элементов своей структуры.

Список литературы

1. Jurafsky D., Martin J. H. *Speech and language processing*. 2nd ed. Boston: Pearson, 2006. 612 p.
2. Manning C. D., Schütze H. *Foundations of statistical natural language processing*. Cambridge: MIT Press, 1999. 696 p.
3. Голубев А. А., Лукашевич Н. В. Исследование моделей нейронных сетей типа BERT для анализа тональности текстов на русском языке // *Науч.-техн. информация. Сер. 2: Информационные процессы и системы*. 2021. № 1. С. 32–41. doi: 10.36535/0548-0027-2021-01-3.
4. Automated end-to-end management of the modeling lifecycle in deep learning / G. Gharibi, V. Walunj, R. Nekkadi, R. Marri, Y. Lee // *Empirical Software Engin.* 2021. Vol. 26. P. 17. doi: 10.1007/s10664-020-09894-9.
5. Маслова М. А., Дмитриев А. С., Холкин Д. О. Методы распознавания именованных сущностей в русском языке // *Инженерный вестн. Дона*. 2021. № 7 (79). С. 93–105.
6. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy / Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, V. Galanos, P. V. Ilavarasan, M. Janssen, P. Jones, A. K. Kar, H. Kizgin, B. Kronemann, B. Lal, B. Lucini, R. Medaglia, K. Le Meunier-FitzHugh, L. C. Le Meunier-FitzHugh, S. Misra, E. Mogaji, S. K. Sharma, J. B. Singh, V. Raghavan, R. Raman, N. P. Rana, S. Samothrakis, J. Spencer, K. Tamilmani, A. Tubadji, P. Walton, M. D. Williams // *Intern. J. of Information Management*. 2019. Vol. 57. P. 1–47. doi: 10.1016/j.ijinfomgt.2019.08.002.
7. Cleveland W. S. Data science: An action plan for expanding the technical areas of the field of statistics // *Statistical Analysis and Data Mining: The ASA Data Sci. J.* 2014. Vol. 7, no. 6. P. 414–417. doi: 10.1002/sam.11239.
8. Jobin A., Ienca M., Vayena E. The global landscape of AI ethics guidelines // *Nature Machine Intelligence*. 2019. Vol. 1, no. 9. P. 389–399. doi: 10.1038/s42256-019-0088-2.
9. The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions / A. F. Borges, F. J. Laurindo, M. M. Spínola, R. F. Gonçalves, C. A. Mattos // *Intern. J. of Information Management*. 2020. Vol. 57. P. 102225. doi: 10.1016/j.ijinfomgt.2020.102225.
10. Robotic process automation: Contemporary themes and challenges / R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. J. Leemans, C. Ouyang, A. H. M. ter Hofstede, I. van de Weerd, M. T. Wynn, H. A. Reijers // *Comp. in Industry*. 2020. Vol. 115. P. 103162. doi: 10.1016/j.compind.2019.103162.

Информация об авторе

Глушенко Артём Геннадьевич – аспирант, старший преподаватель кафедры Информационные системы СПбГЭТУ «ЛЭТИ».
E-mail: artemglushenko98@gmail.com

References

1. Jurafsky D., Martin J. H. *Speech and language processing*. 2nd ed. Boston: Pearson, 2006. 612 p.
2. Manning C. D., Schütze H. *Foundations of statistical natural language processing*. Cambridge: MIT Press, 1999. 696 p.
3. Golubev A. A., Lukashevich N. V. Issledovanie modelej nejronnyh setej tipa BERT dlja analiza tonal'nosti tekstov na russkom jazyke // *Nauch.-tehn. informacija. Ser. 2: Informacionnye processy i sistemy*. 2021. № 1. S. 32–41. doi: 10.36535/0548-0027-2021-01-3. (In Russ.).
4. Automated end-to-end management of the modeling lifecycle in deep learning / G. Gharibi, V. Walunj, R. Nekkadi, R. Marri, Y. Lee // *Empirical Software Engin.* 2021. Vol. 26. P. 17. doi: 10.1007/s10664-020-09894-9.
5. Maslova M. A., Dmitriev A. S., Holkin D. O. Metody raspoznavanija imenovannyh sushhnostej v russkom jazyke // *Inzhenernyj vestn. Dona*. 2021. № 7 (79). S. 93–105. (In Russ.).
6. Artificial Intelligence (AI): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy / Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, V. Galanos, P. V. Ilavarasan, M. Janssen, P. Jones, A. K. Kar, H. Kizgin, B. Kronemann, B. Lal, B. Lucini, R. Medaglia, K. Le Meunier-FitzHugh, L. C. Le Meunier-FitzHugh, S. Misra, E. Mogaji, S. K. Sharma, J. B. Singh, V. Raghavan, R. Raman, N. P. Rana, S. Samothrakis, J. Spencer, K. Tamilmani, A. Tubadji, P. Walton, M. D. Williams // *Intern. J. of Information Management*. 2019. Vol. 57. P. 1–47. doi: 10.1016/j.ijinfomgt.2019.08.002.
7. Cleveland W. S. Data science: An action plan for expanding the technical areas of the field of statistics // *Statistical Analysis and Data Mining: The ASA Data Sci. J.* 2014. Vol. 7, no. 6. P. 414–417. doi: 10.1002/sam.11239.
8. Jobin A., Ienca M., Vayena E. The global landscape of AI ethics guidelines // *Nature Machine Intelligence*. 2019. Vol. 1, no. 9. P. 389–399. doi: 10.1038/s42256-019-0088-2.
9. The strategic use of artificial intelligence in the digital era: Systematic literature review and future research directions / A. F. Borges, F. J. Laurindo, M. M. Spínola, R. F. Gonçalves, C. A. Mattos // *Intern. J. of Information Management*. 2020. Vol. 57. P. 102225. doi: 10.1016/j.ijinfomgt.2020.102225.

10. Robotic process automation: Contemporary themes and challenges / R. Syed, S. Suriadi, M. Adams, W. Bandara, S. J. J. Leemans, C. Ouyang, A. H. M. ter Hofstede, I. van de Weerd, M. T. Wynn, H. A. Reijers // *Comp. in Industry*. 2020. Vol. 115. P. 103162. doi: 10.1016/j.compind.2019.103162.

Information about the author

Artyom G. Glushchenko – postgraduate student, senior lecturer of the Department of Information Systems of Saint Petersburg Electrotechnical University.

E-mail: artemglushenko98@gmail.com

Статья поступила в редакцию 02.12.2025; принята к публикации после рецензирования 29.12.2025; опубликована онлайн 26.02.2026.

Submitted 02.12.2025; accepted 29.12.2025; published online 26.02.2026.
