УТВЕРЖДЕНО
52457223.425000.567-01 90 01-ЛУ

Файл содержит пример оформления технического задания (ТЗ) на доработку интеллектуального модуля ORFO — знакомого каждому инструмента в Word, который исправляет опечатки и грамматические ошибки.
Описаны задачи по обновлению словарей, улучшению обработки букв «е» и «ё», ускорению генерации данных и модернизации архитектуры системы. В результате этих изменений привычный помощник станет ещё умнее, быстрее и точнее.

Подп. и дата | Инв. № дубл. | Взам. инв. № | Подп. и дата | Инв. № подл.

ДОРАБОТКА СИСТЕМЫ АВТОМАТИЧЕСКОГО ПОИСКА И КОРРЕКЦИИ ОШИБОК В ТЕКСТАХ НА ЕСТЕСТВЕННОМ ЯЗЫКЕ

«ОРФО AI»

Техническое задание
52457223.425000.567-01 90 01

Листов 18

2023

Литера **У**

АННОТАЦИЯ

Настоящее техническое задание описывает доработку программного обеспечения ОРФО AI — системы автоматического поиска и коррекции ошибок в текстах на естественном языке. Работы выполняются за счёт собственных средств ООО «Нейросети Ашманова» на основании приказа Генерального директора. Целью доработки является повышение качества, реалистичности и полноты обучающих данных, что необходимо для удовлетворения новых требований лингвистов и повышения точности работы нейросетевых моделей. В рамках доработки планируется:

- реализация поддержки обработки взаимозаменяемых букв "е" и "ё";
- расширение функциональности генерации обучающих данных за счёт внедрения мультипроцессинга и логгирования;
- валидация словарей пар слов и словосочетаний;
- модернизация автоматического валидатора;
- создание интеграционного компонента в модуле генерации обучающих корпусов.

Доработка предполагает развитие модульной архитектуры системы и направлена на устранение ранее выявленных ограничений, повышение устойчивости и масштабируемости решения. В документе приведены цели, задачи, функциональные и технические требования, а также порядок тестирования и приёмки результатов работ.

СОДЕРЖАНИЕ

1 ВВЕДЕНИЕ

Программный комплекс ОРФО AI представляет собой систему автоматического поиска и коррекции ошибок в текстах на естественном языке. Инициатором проекта является ООО «Нейросети Ашманова». При создании используются достижения в области компьютерной лингвистики, обработки естественного языка и нейросетевых технологий.

ОРФО AI построена на модульной архитектуре, сочетающей эвристические и нейросетевые методы анализа. Такая структура позволяет выявлять широкий спектр языковых ошибок, включая сложные случаи: слитное/раздельное написание, формы на -ться/-тся, паронимы, стилистические отклонения и др.

В состав системы входят:

- центральный управляющий модуль, координирующий взаимодействие компонентов;
- эвристический модуль, реализующий правила поиска грамматических и стилистических ошибок;
- нейросетевой модуль, основанный на трансформерных моделях (BERT);
- вспомогательные модули генерации и обучения, обеспечивающие создание синтетических корпусов на основе словарей и параметров конфигурации.

Система обеспечивает работу как на CPU, так и на GPU, демонстрируя высокую производительность при обработке больших объёмов данных.

Программное обеспечение ОРФО AI возможно интегрировать в сторонние информационные системы для последующего применения в следующих сценариях:

- в составе программных продуктов линейки ОРФО, предназначенных для автоматизированной проверки текстов;
- в офисных приложениях, включая Microsoft Word, Microsoft Outlook, LibreOffice Writer;
- в корпоративных и образовательных платформах, ориентированных на анализ письменных работ и формирование экзаменационных материалов;
- в составе интеллектуальных ассистентов, чат-ботов и иных диалоговых решений.

Допускается реализация иных сценариев интеграции и использования, в зависимости от требований Заказчика и архитектуры целевой системы.

Настоящее техническое задание содержит требования к доработке системы ОРФО AI, осуществляемой за счёт собственных средств ООО «Нейросети Ашманова» на основании приказа Генерального директора. Цель доработки — повышение качества, полноты и реалистичности

синтетических обучающих данных, устранение выявленных недостатков, а также расширение функциональности за счёт развития модульной архитектуры и внедрения поддержки мультипроцессной обработки.

Доработка обусловлена необходимостью повышения качества и полноты обучающих данных, применяемых при обучении нейросетевых моделей.

Планируется:

- реализация поддержки новых требований лингвистов;
- устранение выявленных недостатков в алгоритмах генерации;
- расширение функциональности за счёт модульной архитектуры и поддержки мультипроцессинга.

2 ЦЕЛЬ И ЗАДАЧИ ДОРАБОТКИ

2.1 Цель доработки

Целью доработки программного обеспечения ОРФО AI является повышение точности, полноты и масштабируемости автоматического поиска и коррекции ошибок в текстах на естественном языке.

Для этого предусматривается:

- усовершенствование механизмов генерации обучающих корпусов синтетических ошибок;
- повышение качества и лингвистической обоснованности синтетических данных;
- внедрение модульной архитектуры с поддержкой многопроцессной обработки;
- обеспечение эффективного взаимодействия и двустороннего обмена данными между модулями системы.

Доработка осуществляется за счёт собственных средств ООО «Нейросети Ашманова» на основании Приказа Генерального директора.

2.2 Назначение доработки

Развиваемая система предназначена для автоматизированной генерации синтетических корпусов текстов с ошибками, применяемых при обучении моделей обработки естественного языка на русском языке для задачи автоматической коррекции ошибок.

Модуль генерации синтетических корпусов представляет собой вспомогательный компонент, разрабатываемый в рамках внутренних задач, связанных с обучением и тестированием нейросетевой модели ОРФО AI. Указанный модуль не предназначен для внешнего распространения и самостоятельного использования. Его функциональность ориентирована на формирование обучающих и тестовых корпусов, необходимых для повышения качества и устойчивости алгоритмов обработки естественного языка.

Предполагаемые области применения сгенерированных корпусов:

- обучение и тестирование моделей обработки естественного языка (NLP);

- выполнение задач лингвистического анализа и исследовательских проектов в области искусственного интеллекта;
- разработка интеллектуальных ассистентов, чат-ботов и систем автоматической проверки текста.

Целевая аудитория синтетических корпусов:

- лингвисты;
- специалисты в области обработки естественного языка и машинного обучения;
- разработчики ИИ-систем и инженеры по созданию обучающих выборок.

2.3 Задачи доработки

В рамках доработки должны быть решены следующие задачи:

- поддержка мультитокенных и дефисных типов ошибок;
- расширение и актуализация словарей ошибок по требованиям лингвистов;
- обработка взаимозаменяемости букв «е» и «ё»;
- развитие валидаторов словарей ошибок;
- внедрение мультипроцессной генерации с параметрической настройкой и логгированием;
- интеграция разработанных компонентов в единую систему с возможностью масштабирования и подключения новых компонентов;
- обеспечение фильтрации и агрегирования результатов генерации;

2.4 Ожидаемые результаты

В результате доработки будет получена усовершенствованная версия модуля генерации обучающего корпуса для нейросетевых моделей ОРФО AI с расширенной функциональностью и улучшенными характеристиками. Система обеспечит:

- генерацию синтетических корпусов с разнообразными типами ошибок: грамматическими, лексическими, стилистическими, дефисными, мультитокенными, включая поддержание соответствия требованиям лингвистов;
- модульную архитектуру с поддержкой масштабирования, независимой разработки и тестирования компонентов.

3 ОСНОВАНИЯ ДЛЯ ДОРАБОТКИ

Основанием для выполнения доработки программного обеспечения ОРФО AI является решение Генерального директора ООО «Нейросети Ашманова», оформленное соответствующим приказом. Доработка осуществляется за счёт собственных средств организации и обусловлена необходимостью повышения качества и полноты синтетических обучающих корпусов, используемых для обучения и валидации нейросетевых моделей обработки естественного языка.

Необходимость доработки вызвана следующими факторами:

- расширение требований со стороны лингвистических специалистов к типам поддерживаемых ошибок;
- необходимость устранения выявленных недостатков в механизмах генерации и валидации обучающих данных;
- потребность в обеспечении модульности, масштабируемости и отказоустойчивости компонентов системы;
- необходимость поддержки новых форм синтетических ошибок, включая мультитокенные конструкции и ошибки на уровне лексики и грамматики;
- обеспечение параллельной и более производительной генерации обучающих данных для масштабирования процессов обучения нейросетевых моделей.

Реализация доработки позволит повысить достоверность, точность и устойчивость функционирования системы ОРФО AI при выполнении задач автоматической проверки текстов на естественном языке.

4 ТРЕБОВАНИЯ К ДОРАБОТКЕ ПРОГРАММЫ ИЛИ ПРОГРАММНОГО ИЗДЕЛИЯ

4.1 Требования по архитектурным изменениям и рефакторингу

4.1.1 Реализовать в валидаторе механизм проверки, позволяющий выявлять пары слов, различающиеся исключительно символами «е» и «ё».

4.1.2 Дополнить класс `DictLoader` функциональностью автоматического формирования дополнительных словарных пар, в которых«ё» заменяется на «е» при сохранении смысла.

4.1.3 Реализовать функциональность в модуле генерации, позволяющую при обработке слов заменять символ «ё»на «е» с сохранением корректной семантики и морфологии слов.

4.1.4 Добавить в модуль генерации поддержку учёта кавычек, знаков препинания и иных непечатаемых или примыкающих символов, неотделимых от искомых слов.

4.1.5 Расширить алгоритм генерации дефисных ошибок: обеспечить возможность порождения ошибок как с удалением, так и с добавлением дефисов (в обе стороны).

4.1.6 Реализовать в модуле генерации поддержку мультитокенных ошибок, охватывающих более одного токена в пределах одной конструкции.

4.1.7 Обновить валидатор словаря пар слов таким образом, чтобы он автоматически обрабатывал и исключал некорректные пары, выявленные при тестировании обновлённой версии словаря.

4.1.8 Разработать и внедрить интеграционный компонент, объединяющий:

4.1.8.1 подсистему поиска предложений-кандидатов для включения в обучающий корпус;
4.1.8.2 механизм сбалансированного сэмплирования по типам синтетических ошибок;
4.1.8.3 функциональность финальной сборки обучающего датасета в требуемом формате.

4.1.9 Перенести все классы и структуры данных, относящиеся к процессу генерации, в соответствующий модуль генерации, обеспечив логическую изоляцию и модульность.

4.1.10 Реализовать механизм идентификации предложений-кандидатов с использованием хэшей, заменив текущую систему идентификации по индексам.

4.2 Требования по оптимизации производительности

4.2.1 Реализовать предварительную индексацию коллекции предложений-кандидатов с целью ускорения поиска и повышения производительности при генерации обучающих данных.

4.3 Требования по исправлению ошибок и улучшению стабильности

4.3.1 Исправить ошибки в модуле генерации, связанные с некорректной обработкой капитализированных (с заглавной буквы) слов.

4.3.2 Перестроить систему логгирования, установив адекватный уровень детализации сообщений ($DEBUG, INFO, WARNING, ERROR$) и исключив избыточные сообщения.

4.3.3 Провести линтинг и рефакторинг новых компонентов кода, обеспечив соответствие требованиям стиля и корректность исполнения.

4.4 Требования по обновлению зависимостей и совместимости

4.4.1 Обеспечить в модуле инференса корректную обработку новых типов синтетических ошибок, внедрённых в рамках доработки.

4.4.2 Обновить версию словаря существующих словоформ русского языка, используемого в системе.

4.4.3 Обновить версию словаря пар слов и словосочетаний, применяемого при генерации ошибок.

4.4.4 Проверить и актуализировать все файлы `requirements.txt` в пакетах системы, зафиксировав используемые версии библиотек.

4.5 Требования по тестированию и валидации

4.5.1 Разработка и реализация тестовых сценариев для категорий ошибок: дефисные, мультитокенные, ошибки капитализации, замены "е/ё", комбинированные случаи;

4.5.2 Проверка словаря пар слов на предмет дубликатов, логических ошибок и некорректных пар;

4.5.3 Сравнение результатов генерации новых типов ошибок с эталонной версией по согласованной методике и выборке.

4.6 Требования к исходным данным и окружению

4.6.1 Для выполнения доработки и проведения валидации необходимо наличие и использование следующих данных и инструментов:

– словари существующих словоформ русского языка (в актуальной версии);
– словари пар слов и словосочетаний, используемых при генерации ошибок, валидации и инференсе;
– корпуса предложений, прошедших предварительную нормализацию (без гипертекстовой и служебной разметки);
– эталонные корпуса с экспертной разметкой ошибок для проведения валидации;
– YAML/JSON-конфигурации всех компонентов (генерации, валидации, инференса);
– библиотеки: Python ≥ 3.7, PyTorch, HuggingFace Transformers, XML/JSON-парсеры и другие, указанные в requirements.txt;
– доступ к вычислительным ресурсам: CPU (Intel/AMD), GPU (опционально), SSD-диск, ОЗУ от 16 ГБ (рекомендуется 64 ГБ).

4.7 Требования к словарям ошибок и структуре обучающих данных

4.7.1 Словари ошибок должны быть реализованы в формате CSV или с обязательными полями:

- `source` — исходное слово/словосочетание;
- `target` — корректная замена;
- `tag` — тип ошибки;
- `direction` — тип генерации (однонаправленная/двунаправленная замена/вставка/удаление).

4.7.2 Обучающие датасеты формируются в формате, совместимом с алгоритмами токенизации текущих нейросетевых моделей ОРФО AI и включают пары "токен – тег". Метки тегов соответствуют структуре словаря ошибок.

5 ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

5.1 Аппаратное обеспечение

Разработка и доработка системы выполняются с использованием вычислительных ресурсов, принадлежащих ООО «Нейросети Ашманова», в соответствии с Приказом Генерального директора.

В рамках доработки предусмотрено обеспечение повышения качества и полноты обучающих данных, устранение выявленных недостатков в алгоритмах генерации, а также расширение функциональных возможностей программного решения за счёт применения модульной архитектуры и поддержки многопроцессорной обработки.

Для выполнения задач разработки используются персональные компьютеры. Генерация синтетических корпусов на больших объёмах данных осуществляется на выделенном сервере с параметрами, соответствующими следующим требованиям:

- компьютеры, оснащённые процессорами Intel или AMD, а также графическими ускорителями NVIDIA или AMD, обеспечивающими уровень вычислительной мощности не ниже среднего;
- объём оперативной памяти: не менее 64 ГБ;
- спользование твердотельных накопителей (SSD) для повышения скорости обработки и генерации текстовых данных.

5.2 Программное обеспечение

Для обеспечения корректной работы и совместимости компонентов системы используются следующие программные средства:

- Операционные системы: Linux-дистрибутивы (например, Ubuntu, CentOS и аналогичные).
- Язык программирования Python версии 3.7 и выше.
- Программные библиотеки и платформы для машинного обучения, включая предобученные нейросетевые модели (BERT, RoBERTa, GPT-2 и аналоги).
- Промежуточные интерфейсы (API), реализованные в формате JSON и XML, для взаимодействия между модулями.
- Средства морфологического и синтаксического анализа, включая сторонние решения, поддерживающие интеграцию с Python.

6 ПОРЯДОК КОНТРОЛЯ И ПРИЁМКИ

6.1 Контроль выполнения доработки

В процессе выполнения доработки необходимо провести следующие виды испытаний и проверок:

1) Модульное тестирование компонентов модуля генерации обучающих данных, направленное на проверку корректности работы отдельных функциональных блоков:
   - подсистемы поиска предложений-кандидатов для включения в обучающий корпус;
   - механизма сбалансированного сэмплирования по типам синтетических ошибок;
   - финальной сборки обучающего датасета в требуемом формате.

2) Интеграционное тестирование для оценки взаимодействия всех блоков модуля генерации, включая корректную передачу данных между ними в форматах совместимость интерфейсов, фильтрацию и разрешение конфликтов между результатами.

6.2 Приёмка доработки

Приёмка выполненной доработки осуществляется по акту внутренней приёмки, составленному по результатам демонстрации работы доработанного программного обеспечения ОРФО AI в условиях, приближённых к реальной эксплуатации.

Процедура приёмки включает:

1) Проведение демонстрации работоспособности обновлённых модулей системы ОРФО AI с использованием тестовых корпусов, подтверждающих выполнение всех функциональных требований, изложенных в настоящем техническом задании.

2) Составление протокола демонстрации, содержащего описание проведённых операций, зафиксированные результаты и выводы о соответствии доработки установленным требованиям.

3) Оформление акта внутренней приёмки выполненных работ по унифицированной форме (например, Акт о приёмке выполненных работ или Акт технической приёмки программного обеспечения).

Акт приёмки подписывается:

- Техническим руководителем проекта (лицом, ответственным за реализацию доработки в рамках подразделения разработки) — как подтверждение технической состоятельности и завершённости работ;
- Руководителем подразделения – заказчика разработки (например, руководителем отдела исследований, продуктовой команды или внутреннего инициатора проекта) — как лицом, принимающим результат;
- Генеральным директором или иным уполномоченным лицом — как утвердившим результат и направившим распоряжение на выполнение работ (на основании Приказа);
- Главным бухгалтером — для регистрации акта в бухгалтерской системе и отражения затрат в финансовом учёте.

Акт приёмки подлежит передаче в бухгалтерию для включения в комплект отчётных документов, подтверждающих целевое использование средств организации.

6.3 Ограничения и риски

В рамках доработки необходимо учитывать следующие ограничения:

- Обработка предложений длиной не более 512 токенов;
- Нагрузочное тестирование проводится на корпусах до 100 000 предложений;
- Наличие дубликатов, некорректных пар в словаре может привести к снижению точности модели;
- Несогласованные версии словарей и конфигураций могут вызывать ошибки исполнения;
- Обработка новых типов ошибок требует дополнительной оперативной памяти и может увеличить время генерации на 15-30%;
- Для стабильной работы рекомендуется использование SSD и выделенной RAM не менее 64 ГБ.

6.4 Требования к квалификации исполнителей

Исполнители доработки должны обладать квалификацией, соответствующей следующим требованиям:

- опыт разработки на языке программирования Python, в том числе в средах с использованием transformers, PyTorch, HuggingFace;

- практические навыки в области машинного обучения;
- понимание принципов модульной архитектуры программных систем и взаимодействия между модулями;
- желателен опыт взаимодействия с лингвистами и разработки алгоритмов обработки естественного языка (NLP).

7 СТАДИИ И ЭТАПЫ ДОРАБОТКИ

7.1 Стадии доработки

В таблице 1 приведены основные стадии доработки системы ОРФО AI, включая содержание работ, ориентировочные сроки и ожидаемые результаты.

Таблица 1 – Основные стадии доработки системы ОРФО AI

№	Стадия доработки	Сроки	Результат
1	Анализ требований	Сентябрь 2023	Формализованная спецификация требований
2	Проектирование архитектуры	Октябрь – Ноябрь 2023	Обновлённая архитектура модулей генерации и валидации
3	Разработка и тестирование	Декабрь 2023 – Декабрь 2024	Рабочий прототип системы, подтверждённый тестированием

7.2 Этапы доработки

В таблице 2 приведены этапы доработки , даты начала этапа и описание работ.

Таблица 2 – Этапы доработки

№	Этап доработки	Начало этапа	Описание работ
1	Начало доработки	01 сентября 2023 года	Запуск проекта, уточнение требований, подготовка инфраструктуры
2	Завершение доработки	31 декабря 2024 года	Завершение реализации, тестирования

8 ПЕРЕЧЕНЬ ТЕРМИНОВ И ПРИНЯТЫХ СОКРАЩЕНИЙ

Термин / Сокращение	Определение
BERT	Bidirectional Encoder Representations from Transformers — одна из трансформерных моделей, используемая для анализа текста
config / конфигурация	Настройки компонентов системы, определяющие параметры генерации, валидации и логгирования
hyphen error / дефисная ошибка	Ошибка, связанная с неправильным использованием дефиса в словах или словосочетаниях
GPT-2	Generative Pretrained Transformer 2 — трансформерная модель для генерации и исправления текста
hash	Хэш-сумма, используемая для идентификации объектов
inference / инференс	Процесс применения обученной нейросетевой модели к новым данным
json	Формат обмена данными JavaScript Object Notation
linting / линтинг	Процедура проверки исходного кода на соответствие стилевым требованиям
log	Журнал событий, используемый для анализа выполнения программы
logging	Процесс ведения логов
multiprocessing	Модель параллельной обработки с использованием нескольких процессов
NER	Named Entity Recognition — задача распознавания именных сущностей
NLP	Natural Language Processing — обработка естественного языка
pair dictionary	Словарь пар слов и словосочетаний, используемый при генерации ошибок
pipeline	Последовательность этапов обработки данных
PyTorch	Фреймворк машинного обучения, используемый для создания и обучения нейросетей
Рефакторинг	Процесс улучшения структуры, качества и читаемости кода без изменения его функциональности
RoBERTa	Robustly Optimized BERT Approach — модификация модели BERT
sampling	Процесс случайного выбора элементов из множества
spaCy	Библиотека Python для NLP
synthetic error	Искусственно сгенерированная ошибка (грамматическая, орфографическая и др.)
token	Минимальная единица текста, обрабатываемая NLP-моделью
tokenizer	Модуль разбиения текста на токены
unicode	Стандарт кодирования символов
validation	Процесс проверки корректности данных или функционала
validator / валидатор	Модуль, осуществляющий проверку сгенерированных данных на корректность
yaml	Формат конфигурационных файлов
генератор ошибок	Подсистема для создания синтетических ошибок
ё/е	Символы кириллицы, обрабатываемые как взаимозаменяемые в контексте генерации
интеграционный модуль	Компонент для объединения подмодулей

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопроводительного документа и дата	Подпись	Дата
	изме-ненных	заме-ненных	новых	аннули-рованных					