

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка модуля предварительной оценки ответов для системы анализа
структурированных текстовых отчетов»
по дисциплине «Проектный практикум»

Заказчик: Филимонов А.Ю.

Куратор: Аксенов К.А.

кандидат технологических наук, доцент

Студент команды 21/ЛКП-5138-2025

Кистанова М.М.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Постановка задачи и анализ решений.....	4
1.1 Требования заказчика	4
1.1.1 Функциональные требования	4
1.1.2 Нефункциональные требования	5
1.2 Требования пользователей	5
1.3 Анализ аналогов	6
1.4 План действий	8
2 Архитектура программного продукта.....	10
2.1 Выбор архитектурного решения.....	10
2.2 Основные компоненты модуля	11
2.3 Описание работы модуля	12
2.3.1 Проверка фиксированных ответов	12
2.3.2 Проверка параметризованных ответов	13
2.3.3 Проверка описательных ответов	14
3 Реализация и тестирование системы.....	16
3.1 Методология разработки	16
3.2 Процесс разработки	16
3.3 Результаты тестирования	17
4 Индивидуальный отчет.....	19
4.1 Кистанова Марина Михайловна	19
ЗАКЛЮЧЕНИЕ	22
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	23

ВВЕДЕНИЕ

В рамках учебного процесса результаты лабораторных работ фиксируются в отчетах посредством системы, интегрированной с LMS Moodle. Несмотря на автоматизацию структурного анализа и подсчета баллов, оценка содержательной части ответов на открытые вопросы по-прежнему требует ручной проверки преподавателем. Ручная проверка характеризуется значительными временными затратами, потенциальной субъективностью оценок и задержками обратной связи.

Существующие инструменты эффективно обрабатывают формализованные данные, однако не обеспечивают полноценного анализа смыслового содержания ответов, сохраняя зависимость от экспертной оценки. Рост учебной нагрузки усугубляет эти сложности, снижая общую эффективность образовательного процесса.

В качестве решения предлагается разработка модуля предварительной оценки ответов, интегрированного в существующую систему анализа отчетов LMS Moodle. Модуль позволит снизить трудозатраты на проверку открытых вопросов, обеспечить своевременную обратную связь обучающимся и предоставить возможность последующей ручной корректировки оценок.

Целью работы является создание модуля автоматизированной проверки различных типов ответов, который генерирует рекомендательные оценки на основе эталонных ответов преподавателя.

Для достижения цели были поставлены следующие задачи:

- определить форматы описания ответов и критерии их оценки,
- реализовать алгоритмы проверки для различных типов ответов,
- доработать интерфейс для возможности задания преподавателем эталонных ответов и корректировки оценок,
- интегрировать модуль с существующей системой анализа отчетов.

1 Постановка задачи и анализ решений

1.1 Требования заказчика

Основная цель заказчика заключается в создании модуля автоматизированной предварительной проверки текстовых ответов студентов в отчетах лабораторных работ. Система должна обеспечивать классификацию ответов на три типа – фиксированные, параметризированные и описательные, с учетом специфики их верификации. Результаты такой проверки не являются окончательными и предназначаются в качестве вспомогательного инструмента преподавателю.

Для реализации цели необходимо определить два типа требований: функциональные, описывающие задачи системы и взаимодействие с пользователями, и нефункциональные, определяющие параметры работы системы, включая стандарты качества, производительности, удобства использования и другие параметры, обеспечивающие эффективность и надежность работы.

1.1.1 Функциональные требования

К основным функциональным требованиям относятся:

- возможность преподавателю указывать тип каждого ответа в шаблоне отчета при редактировании ответа;
- возможность задавать эталонный ответ соответствующего типа: фиксированное значение, шаблон с параметрами или перечень ключевых тезисов;
- автоматическая проверка ответов, инициируемая при получении отчета от студента;
- формирование для каждого ответа рекомендуемой оценки и комментария на основе сопоставления с эталоном;
- предоставление преподавателям и ассистентам доступа к результатам автоматической проверки через интерфейс;

- возможность преподавателю редактировать автоматически сформированные оценки и комментарии до их передачи в LMS.

1.1.2 Нефункциональные требования

К основным нефункциональным требованиям относятся:

- среднее время выполнения автоматической проверки одного отчета не должно превышать 5 секунд,
- максимальное предельное время выполнения проверки отчета не должно превышать 15 минут с момента его загрузки,
- отчеты проверяются по одному одновременно в порядке FIFO,
- необходимость расширения пользовательского интерфейса существующей системы для преподавателей и ассистентов с целью интеграции модуля предварительной оценки для визуализации результатов проверки.

1.2 Требования пользователей

На основе анализа требований заказчика сформировано общее представление о целях проекта и его ключевых характеристиках. Однако для создания удобной и эффективной системы важно учесть потребности конечных пользователей, которые будут непосредственно взаимодействовать с системой.

Система предназначена для предварительной автоматизированной проверки отчетов, поэтому основные требования сформулированы для преподавателей и ассистентов — непосредственных пользователей системы.

Требования студентов не рассматриваются, так как их роль ограничивается отправкой отчетов, а оценка и обработка результатов выполняется преподавательским составом, без предоставления обучающимся доступа к промежуточным результатам проверки.

Требования преподавателей и ассистентов включают:

- ручное задание и редактирование эталонных ответов с использованием интерфейса системы без обращения к разработчикам,
- просмотр результатов автоматической проверки непосредственно в интерфейсе системы,
- ручное изменение сгенерированных модулем баллов перед отправкой в LMS.

1.3 Анализ аналогов

Для выбора оптимального метода автоматизированной оценки русскоязычных ответов в задачах NLP был проведен детальный сравнительный анализ наиболее распространенных подходов. В ходе исследования мы оценили не только традиционные алгоритмы, но и современные нейросетевые модели, а также гибридные схемы, объединяющие несколько методик сразу. Сравнение выполнялось по следующим ключевым критериям:

- 1) парадигма метода – принцип представления текстовых данных: грамматические правила, семантические векторы, нейросети, графы;
- 2) алгоритм сравнения – способ определения соответствия ответа эталону: например, такие, как экспертная оценка, метрики векторного пространства, машинное обучение и так далее;
- 3) количественные показатели качества – Accuracy и F-мера, характеризующие точность классификации;
- 4) приблизительное время обработки одного ответа – оценка вычислительной сложности на основе реального времени обработки;
- 5) трудоемкость настройки – объем трудозатрат на адаптацию метода к предметной области: низкая (< 8 ч), средняя (8-40 ч), высокая (40-200 ч), очень высокая (> 200 ч).

Результаты анализа представлены в таблице 1.

Таблица 1 – Сравнительные характеристики методов автоматизированной оценки текстовых ответов

Парадигма метода	Алгоритм	Качество	Время обработки	Трудоемкость настройки
Грамматические правила (спец.)	Экспертная оценка	Accuracy < 50%	~0,1–0,3 сек	Высокая
Грамматические правила (унив.)	Доля правильности	F-мера 60-65%	~1.3 сек	Высокая
Семантические векторы (word2vec)	Косинусная метрика	F-мера 55–74%	~0,1–0,3 сек	Средняя
Гибридная (BERT + word2vec)	Случайный лес	Accuracy 77%, F-мера 70%	~1-2 сек	Средняя
Граф синтаксических связей	Косинусная метрика	F-мера 90%	~0,5–3 сек	Очень высокая

Сравнительный анализ существующих методов автоматизированной оценки ответов показывает, что различные подходы имеют разную применимость в учебном процессе.

Методы на основе грамматических правил [1, 2] демонстрируют относительно низкое время обработки, однако требуют значительных усилий по ручной разработке грамматик и словарей. Их точность ограничена, что связано с жесткой зависимостью от правил и ограниченной гибкостью.

Методы на основе семантических векторов [3] предлагают баланс между скоростью и качеством оценки, с умеренной трудоемкостью настройки, включающей разметку и дообучение под предметную область

Гибридные методы с использованием BERT и word2vec [4] показывают улучшенные показатели качества при умеренном времени обработки. Настройка также требует разметки и дообучения, что соответствует средней трудоемкости.

Методы, основанные на графах синтаксических связей [5], достигают наивысшей точности, но характеризуются самой высокой трудоемкостью настройки, которая связана с необходимостью ручной разработки правил, словарей, а также тонкой настройки параметров и порогов.

С увеличением точности методов автоматизированной оценки текстовых ответов возрастают их вычислительные и организационные издержки, что существенно ограничивает возможность внедрения наиболее точных решений в массовых образовательных системах.

В практических условиях выбор подхода определяется необходимостью баланса между качеством оценки, доступными вычислительными ресурсами и трудоемкостью настройки и сопровождения системы. В качестве компромисса предлагается гибридный подход, сочетающий методы различной сложности в зависимости от типа ответа. Однако для его реализации необходима предварительная классификация типов ответов.

1.4 План действий

Для организации работы и достижения поставленной цели необходимо создание плана действий. План станет основным рабочим документом, позволяющим концентрировать усилия на разработке частей системы, отслеживать прогресс и своевременно вносить корректизы и обеспечивающим прозрачность процесса разработки для заказчика и других сторон.

В связи с тем, что разработка осуществляется одним человеком, было принято решение начать разработку только после завершения необходимого анализа. План разбит на две части: аналитика и разработка.

К части аналитики решения относятся следующие задачи:

- проведение анализа ЦА и конкурентов – 5 дней,
- выделение требований для оценки каждого типа ответа – 3 дня,
- разработка формата описания ответов – 4 дня.

К части разработки решения относятся следующие задачи:

- модификация визуализации данных настройки ответов – 2 дня,
- реализация логики оценки фиксированных ответов – 5 дней,
- реализация логики оценки параметризованных ответов – 5 дней,
- реализация логики оценки описательных ответов – 5 дней,

- реализация интерфейса для сохранения эталонных ответов – 2 дня,
- расширение интерфейса преподавателя – 4 дня,
- оценка точности оценок по видам ответов – 5 дней,
- проведение тестирования системы – 7 дней,
- деплой решения – 2 дня.

2 Архитектура программного продукта

2.1 Выбор архитектурного решения

Для разработки модуля предварительной оценки отчетов лабораторных работ была выбрана асинхронная многокомпонентная архитектура. Это решение обусловлено необходимостью эффективного распределения нагрузки при обработке CPU-bound задач, связанных с анализом текста. Механизм предварительной оценки реализован как многоэтапный процесс, начинающийся с получения данных отчета из системы, последующей классификации ответов по типам, сравнения с эталонными решениями и выполнения специализированных проверок для каждого типа.

При отправке отчета на проверку запускается фоновая задача. Для обработки, связанных с анализом содержимого ответов, применяется асинхронный подход, использующий пул процессов на основе ProcessPoolExecutor в Python для параллельного выполнения проверок. Оптимальность использования ресурсов достигается минимальным количеством рабочих процессов, поскольку потребность в мгновенной оценке отсутствует.

Предварительная классификация ответов необходима для эффективной автоматизации проверки. Разные по структуре и содержанию ответы требуют принципиально различных подходов к оценке их правильности. На основе анализа типов ответов в лабораторных работах была разработана классификация и подобраны соответствующие методы верификации, представленные в таблице 2.

Таблица 2 – Сравнительный анализ типов ответов

Тип ответа	Метод верификации	Особенности обработки
Фиксированный	Лексическое совпадение	Учет морфологических и синтаксических вариаций
Параметризованный	Двухэтапная верификация с подстановкой значений	Обработка диапазонов и ссылок на другие ответы
Описательный	Семантический анализ	Выявление ключевых тезисов с учетом перифраза

2.2 Основные компоненты модуля

Модуль предварительной оценки ответов студентов, интегрированный в серверную часть системы анализа лабораторных работ, существенно расширяет ее возможности. В основе модуля лежит сложная логика анализа текста для различных форматов ответов, включая фиксированные, параметризованные и описательные. Этот модуль не является самостоятельной системой, а предназначен для интеграции в существующую инфраструктуру.

Центральным элементом модуля является ядро логики оценки, которое содержит основные алгоритмы верификации. Эти алгоритмы осуществляют сравнение ответа студента с эталоном и формируют результат проверки в виде рекомендуемого балла и текстового комментария. Для каждого типа ответа в ядре реализована специализированная логика.

Особое внимание уделено эффективному управлению вычислительными ресурсами. Некоторые операции, особенно при работе с описательными ответами, могут быть вычислительно затратными. Для предотвращения блокировки основного потока обработки, модуль использует обработчик асинхронных задач. Этот компонент управляет выполнением ресурсоемких операций в фоновом режиме, используя механизм пула процессов (ProcessPoolExecutor). Ядро логики оценки делегирует выполнение этих задач обработчику, который уведомляет о завершении.

Взаимодействие модуля с существующей системой осуществляется через четко определенный программный интерфейс, предоставляемый серверной частью модуля. Именно через этот интерфейс существующая система передает ответы студентов для оценки и получает обратно сгенерированные модулем результаты предварительной проверки.

2.3 Описание работы модуля

2.3.1 Проверка фиксированных ответов

Данный механизм оценки отвечает за проверку ответов фиксированного формата, используя последовательный алгоритм, визуализированный на рисунке 1.

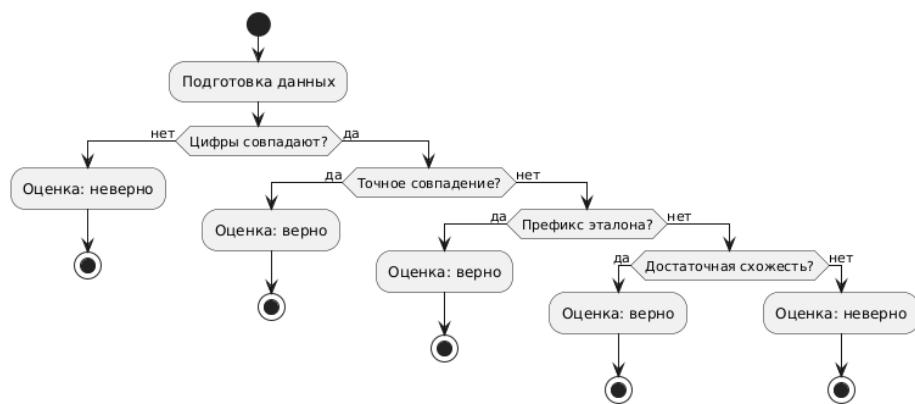


Рисунок 1 – Диаграмма деятельности механизма оценки ответов фиксированного формата

Сначала система проводит прямое сравнение, учитывая регистр символов и пробельные знаки, чтобы подтвердить полное совпадение ответа учащегося с заданной эталонной формулировкой. Если же обнаруживаются расхождения, система переходит к анализу на предмет известных префиксов или распространенных аббревиатур, что помогает учесть возможные вариации в изложении термина.

При отсутствии необходимых уточнений применяется алгоритм нечеткого частичного сравнения, основанном на измененных вариантах метрик редактирования строк. Этот метод позволяет оценить степень схожести между ответом и эталоном даже при наличии незначительных ошибок или опечаток, что существенно повышает гибкость системы.

2.3.2 Проверка параметризованных ответов

При оценке параметризованных ответов модуль опирается на эталонный шаблон, в котором определены необходимые параметры – диапазоны допустимых значений или ссылки на результаты других частей отчета. Как видно из схемы на рисунке 2, процесс начинается с динамической подстановки фактических данных, представленных учащимся, вместо идентификаторов в формате {custom_answer_element_id}.

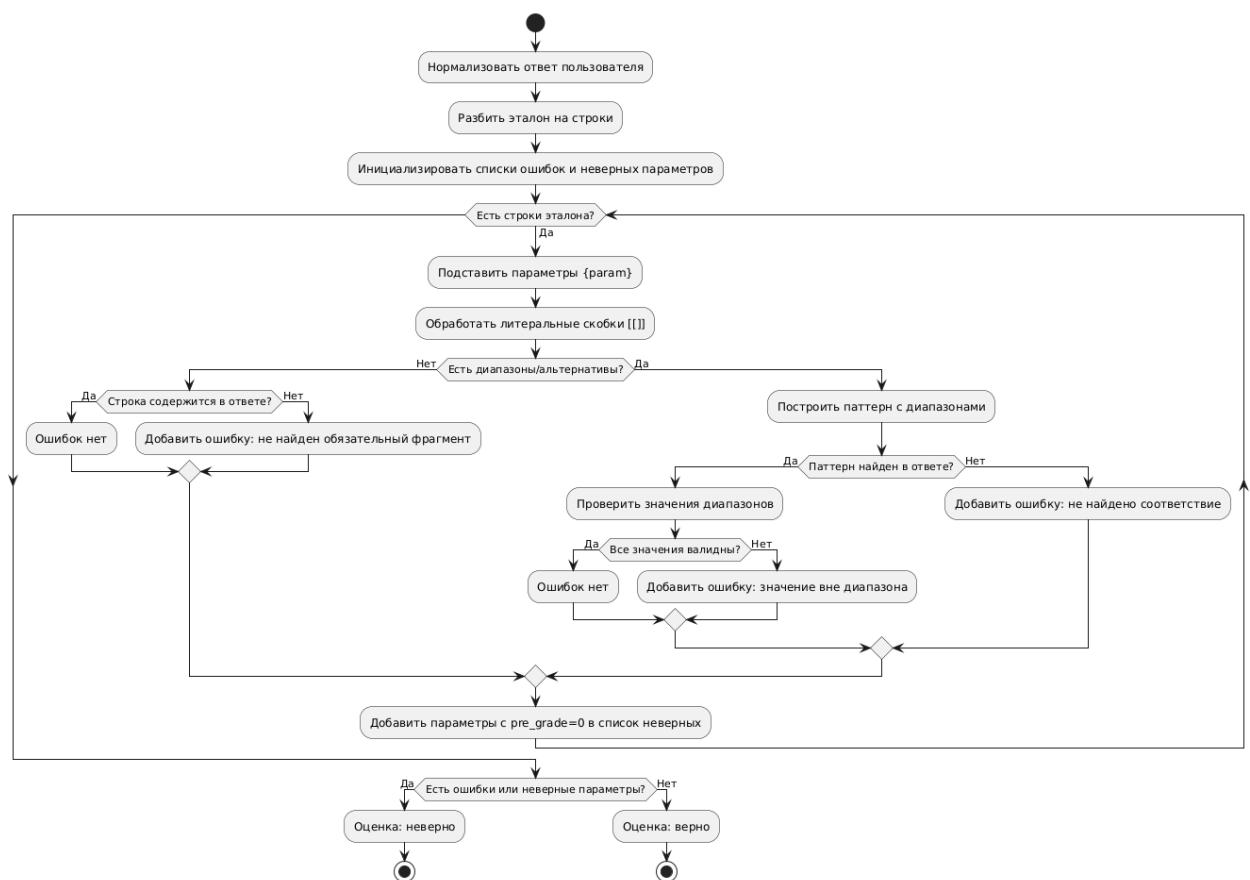


Рисунок 2 – Диаграмма деятельности механизма оценки ответов параметризованного формата

Такой подход позволяет индивидуализировать проверку ответа, поскольку каждое значение или диапазон оцениваются в зависимости от контекста задачи. После подстановки всех параметров формируется итоговая строка, которая сравнивается с ответом учащегося с использованием метода точного

поиска по подстрокам. При этом каждая часть шаблона проходит независимую проверку, а суммарная успешность поиска всех требуемых элементов в ответе определяет окончательную оценку.

2.3.3 Проверка описательных ответов

Оценка описательных ответов реализуется через анализ семантической близости между текстом учащегося и ключевыми тезисами эталонного ответа, процесс которого иллюстрирован на рисунке 3.

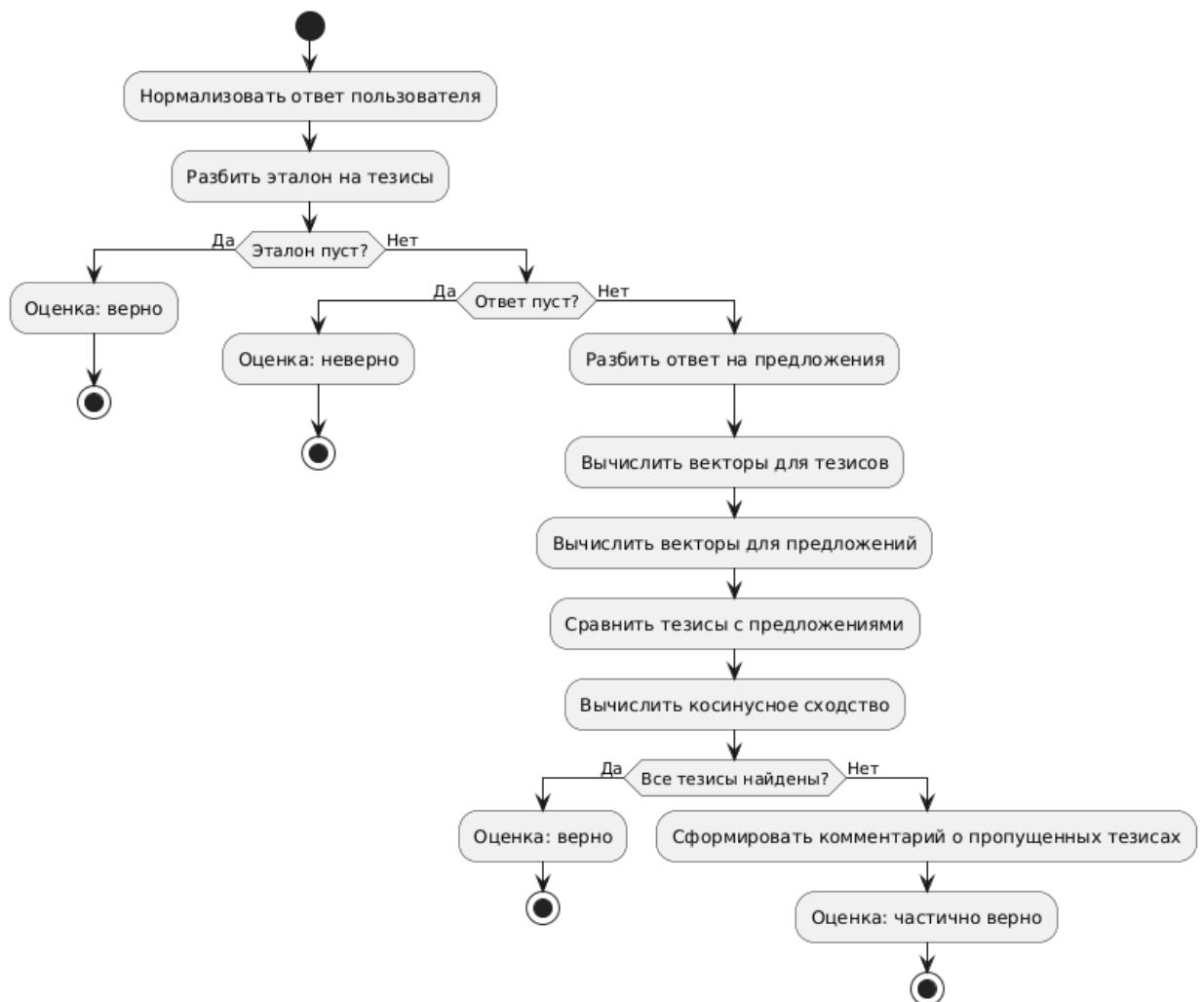


Рисунок 3 – Диаграмма деятельности механизма оценки ответов описательного формата

Вначале как эталонный ответ, так и исходный текст разбиваются на отдельные предложения с помощью регулярных выражений, что позволяет выделить отдельные смысловые единицы.

Каждый тезис эталона затем преобразуется в векторное представление с использованием метода взвешенной суммы эмбеддингов слов, где различные части речи получают разные веса: например, существительные оказываются наиболее значимыми, а глаголы и числительные – несколько менее весомы.

Полученные векторы нормализуются с учетом суммарного веса, что позволяет сопоставлять их посредством косинусного сходства, нормализованного до диапазона от 0 до 1. При достижении значения схожести, превышающего установленные пороги, соответствующий тезис считается успешно обнаруженным в тексте учащегося.

Итоговая оценка определяется по соотношению найденных тезисов к общему количеству элементов, а в случае пропуска некоторых компонентов формируется комментарий с рекомендациями по доработке ответа, что помогает учащемуся скорректировать недочеты.

3 Реализация и тестирование системы

3.1 Методология разработки

Для обеспечения максимальной гибкости и адаптивности в процессе разработки системы, несмотря на то что команда состояла всего из одного человека, была выбрана Agile-методология. Такой подход позволил быстро реагировать на изменяющиеся требования к проекту и обеспечивать наиболее эффективное использование доступных ресурсов.

В контексте данного проекта применение Agile-методологии оказалось особенно полезным, поскольку она способствовала достижению следующих ключевых преимуществ:

- позволяла быстро адаптироваться к изменениям в требованиях и технологиях;
- обеспечивала фокус на доставке рабочего программного обеспечения, а не на соблюдении формальных процессов;
- снизила риск проекта, поскольку ошибки и проблемы были выявлены и исправлены на ранних этапах.

3.2 Процесс разработки

Процесс разработки модуля охватывал как серверную, так и клиентскую части системы. Разработка серверной части осуществлялась на языке Python с использованием фреймворка FastAPI. Что касается клиентской части, для ее создания были задействованы сборщик Vite, библиотека React, CSS фреймворк Tailwind и язык программирования TypeScript. В качестве основной базы данных была выбрана и использовалась MySQL.

Для эффективного управления исходным кодом и обеспечения командной работы применялась система контроля версий Git с хранением кода на

платформе GitHub. В частности, была применена стратегия ветвления feature-branch. Данной подход позволял изолировать разработку новых функций в отдельных ветках, обеспечивая их интеграцию в основную ветку лишь после успешного завершения всех этапов разработки и тщательного прохождения тестирования.

3.3 Результаты тестирования

Тестирование разработанной системы было организовано и проведено в два основных этапа. Первый этап включал проверку корректности работы механизмов оценки, тогда как второй был сфокусирован на проверке отказоустойчивости реализации фоновых задач.

Для проверки механизмов оценки было разработано и выполнено 178 юнит-тестов. Среди них, наиболее простыми в реализации и верификации оказались 14 тестов, предназначенных для проверки описательных ответов. В то же время, наибольшую сложность представляли 125 тестов для параметризованных ответов. Оставшиеся тесты были посвящены проверке фиксированных ответов.

Положительным результатом стало то, что все эти тесты подтвердили корректность реализации механизмов оценки. Это включало успешную работу с различными граничными случаями, такими как ответы, содержащие опечатки, неполные ответы или ответы с избыточной информацией, демонстрируя надежность модуля при работе с большим многообразием ответов.

Что касается второго этапа, тестирование фоновых задач проводилось посредством использования обширного набора модульных тестов, которые были специально разработаны для имитации различных сценариев обработки результатов фоновых процессов. К сожалению, согласно полученным результатам тестирования, были обнаружены определенные недочеты в реализации обработки фоновых задач, как это демонстрирует Рисунок 4.

✓	✖ BackgroundTaskServiceTests	26 ms
✖	test_invalid_data_handling (Проверка 2 ms)	
✖	test_non_list_result_normalization (Проверка 3 ms)	
✖	test_null_result_handling (Проверка 2 ms)	
✖	test_rollback_on_commit_failure (Проверка 2 ms)	
✖	test_rollback_on_save_error (Проверка 2 ms)	
✖	test_rollback_on_task_error (Проверка 2 ms)	

Рисунок 4 – Результаты тестирования реализации обработки фоновых задач

Во-первых, было выявлено, что в ряде тестовых сценариев, где ожидалось возникновение специфических исключений, исключения не генерировались должным образом. Это явно указывает на неполную или недостаточную проверку входных параметров в соответствующих модулях.

Во-вторых, обнаружилась некорректная нормализация результатов, которые не являлись списками. Это приводило к серьезным проблемам, таким как несовпадение ожидаемого формата данных при попытке вызова метода сохранения, что могло вызвать дальнейшие ошибки.

Особое внимание было уделено аспектам управления транзакциями. Было установлено, что при возникновении ошибок в ходе выполнения фоновой задачи, корректный вызов метода отката транзакции не был гарантирован. Действительно, в ряде проведенных тестов операция rollback не выполнялась должным образом, что прямо препятствует корректному и полному восстановлению состояния системы в случае сбоев и нарушает ее целостность.

После выявления этих недочетов, была проведена оперативная и всесторонняя работа по их устранению. Все обнаруженные проблемы были тщательно проанализированы, причины их возникновения установлены, и были реализованы соответствующие исправления.

4 Индивидуальный отчет

4.1 Кистанова Марина Михайловна

Для реализации механизма предварительной оценки ответов в интерфейс преподавателя было добавлено поле для ввода эталонного ответа, формат которого зависит от типа. Настройка ответа включает идентификатор, вес (0-20 баллов), тип и эталонный ответ.

Для параметризованных ответов предусмотрены ссылки на другие элементы в формате {id_элемента}. Например, эталон 10.0.0.{pc_num}, представленный на рисунке 5, указывает, что последний октет IP-адреса должен соответствовать значению из ответа с идентификатором pc_num.

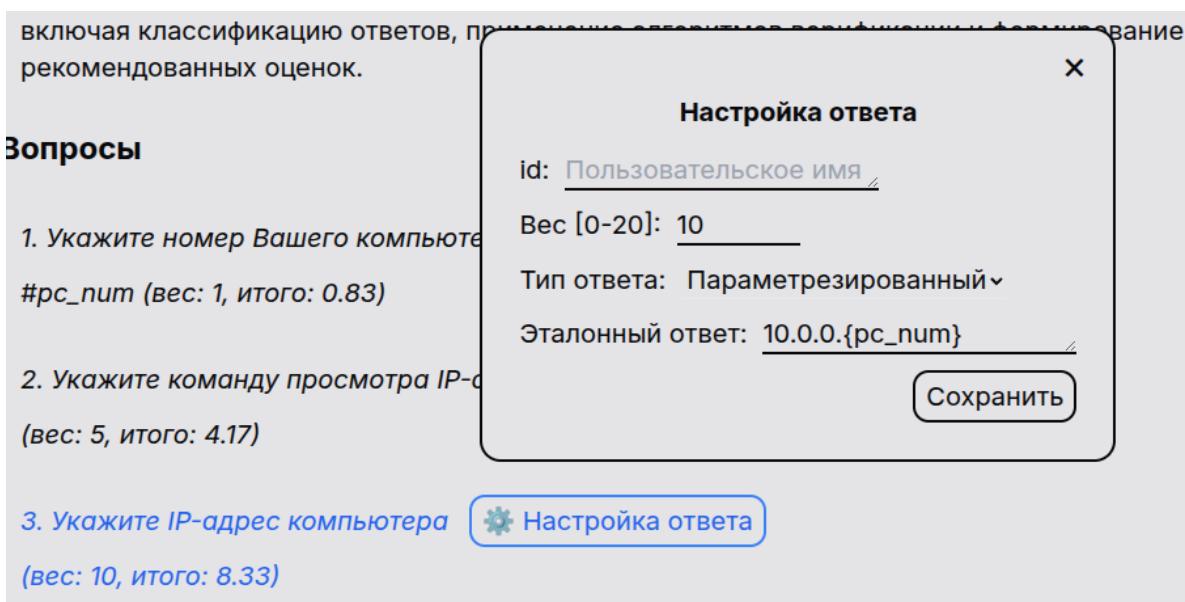


Рисунок 5 – Указание эталонного параметризованного ответа

При проверке отчета система отображает предварительную оценку: верные ответы (>0 баллов) отмечаются галочкой, неверные (0 баллов) - крестом с пояснением причины ошибки (рисунок 6).

1. Укажите номер Вашего компьютера
▼ [Предварительно неверно]
В тезисе '[1-20]' значение '23' не попадает в диапазон [1-20]
23
2. Укажите команду просмотра IP-адреса компьютера
▼ [Предварительно верно]
Достаточное соответствие: текущее - 87.50%, порог - 87.15%
ipconfig

Рисунок 6 – Визуализация результатов проверки фиксированного и описательного ответов

Преподаватель может изменить оценку вручную. Особенностью параметризованных ответов является то, что неверное значение любого параметра делает весь ответ неверным, что демонстрируется на рисунке 7.

3. Укажите IP-адрес компьютера
▼ [Предварительно неверно]
Параметр 'pc_num' предварительно неверен (score=0)
10.0.0.23

Рисунок 7 – Визуализация оценки с неверно оцененным параметром

Для оценки эффективности системы были проведены исследования точности алгоритмов, сравнивающие автоматическую и экспертную оценку. Исследование охватило 164 фиксированных ответа, 290 параметризованных и 28 ответов-рассуждений из 10 лабораторных работ. Результаты показали, что алгоритм проверки фиксированных ответов демонстрирует наивысшую точность - 96,86% благодаря простоте задачи сравнения строк.

Точность алгоритма параметризованных ответов составила 88,89%, что объясняется необходимостью обработки параметров и их зависимостей. Основные сложности возникают при работе с вложенными параметрами и

цепочками зависимостей, где ошибка в одном звене вызывает каскадный эффект.

Алгоритм проверки ответов-рассуждений продемонстрировал точность 66,67% из-за сложности семантического анализа текста и высокой вариативности формулировок студентов. Несмотря на относительно невысокую точность, система эффективно выполняет предварительную фильтрацию, автоматически оценивая простые случаи и выделяя сложные для экспертной оценки преподавателем.

Таким образом, точность реализованных алгоритмов соответствует сложности решаемых задач и является достаточной для практического применения в образовательном процессе при предварительной оценке отчетов лабораторных работ.

ЗАКЛЮЧЕНИЕ

В данной работе рассмотрена проблема автоматизации проверки открытых текстовых ответов в рамках образовательного процесса с использованием LMS Moodle. Анализ существующих решений показал, что современные методы обработки естественного языка демонстрируют широкую вариативность по точности, вычислительной сложности и трудозатратам на внедрение, однако ни один из них в отдельности не обеспечивает оптимального баланса между качеством оценки и ресурсной доступностью для применения в реальной учебной практике.

Для преодоления выявленных ограничений предложен гибридный подход, основанный на предварительной классификации типов ответов и применении соответствующих методов оценки: от лексического сопоставления для кратких ответов до семантического анализа с выделением ключевых тезисов – для развернутых. Разработан формат представления эталонных ответов, включающий двухуровневую модель: интуитивный интерфейс для преподавателя и машиночитаемое представление для системы.

Реализация модуля предварительной оценки, интегриированного в существующую систему анализа отчетов, позволит значительно снизить трудоемкость проверки, минимизировать субъективность оценивания и обеспечить прозрачность оценки. Это, в свою очередь, повысит эффективность учебного процесса и создаст предпосылки для масштабируемости системы при росте числа учащихся.

В дальнейшем планируется разработка прототипа модуля, проведение пилотного тестирования и анализ его точности на основе реальных ответов учащихся, включая возможность генерации отдельной обратной связи учащимся.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Прокопьев Н. А. Автоматизированная оценка ответов при контроле знаний для вопросов типа «Определение» и «Описание» // Ученые записки Казанского университета. Серия: Физико-математические науки. — 2024. — Т. 166, № 4. — С. 580–593.
2. Кожевников В. А., Сабинин О. Ю. Система автоматической проверки ответов на открытые вопросы на русском языке // Научно-технические ведомости СПбГПУ. Информатика. Телекоммуникации. Управление. — 2018. — Т. 11, № 3. — С. 57–72.
3. Миннегалиева С., Сабитова Г., Гаялиев А. Метод предварительной оценки ответов студентов на основе векторной модели документов // Журнал «Российские цифровые библиотеки». — 2023. — Т. 26, № 3. — С. 324–339.
4. Миннегалиева С., Кашапов И., Морозова О. Автоматизированная оценка кратких ответов студентов с использованием языковых моделей // Журнал «Российские цифровые библиотеки». — 2024. — Т. 27, № 3. — С. 278–293.
5. Леонов А. Г., Мартынов Н. С., Машенко К. А., Холкина А. А., Шляхов А. В. Автоматизация семантического анализа текстовых ответов студентов в цифровой образовательной платформе // Программные продукты и системы. — 2024. — Т. 37, № 3. — С. 440–452.