Автоматизация рецензирования отчетов по лабораторным работам с использованием больших языковых моделей

О.Д. Асенчик, В.А. Воробьев
Учреждение образования «Гомельский государственный технический университет имени П.О. Сухого»

В условиях цифровой трансформации высшего технического образования происходит активное внедрение инновационных технологий в учебный процесс с целью повышения его эффективности. Одним из трудоемких аспектов в деятельности преподавателя IT-дисциплин остается проверка отчетов по лабораторным работам. Особенность таких работ заключается в сложности структуры и большом количестве уникальных вариантов заданий. Отчеты студентов по программированию, базам данных или сетевым технологиям содержат не только текстовые описания, но и фрагменты программного кода, таблицы, диаграммы, SQL-запросы и результаты их выполнения. Проверка таких работ требует от преподавателя оценки соответствия формальным требованиям, анализа логики представленных решений, а также корректности синтаксиса и семантики кода, что при большом количестве студентов и вариантов заданий приводит к значительным временным затратам и риску субъективной оценки.

Появление и развитие больших языковых моделей (LLM) открывает новые возможности для автоматизации анализа и оценки текстов и программного кода при его наличии. Современные LLM демонстрируют высокую эффективность анализа контента благодаря их способности к глубокому семантическому разбору сложных технических текстов, обработке структурированных данных (кода, формул) и генерации содержательных оценочных суждений, основанных на выявлении логических взаимосвязей в представленном материале. Важным является возможность LLM выявлять не только очевидные ошибки, но и скрытые проблемы, такие как нарушение принципов проектирования в коде или логические несоответствия в объяснениях студентов, что значительно превышает возможности традиционных автоматизированных систем проверки. LLM демонстрируют высокую способность к пониманию контекста и выявлению логических связей в текстах [1], что делает их особенно подходящими для оценки работ, содержащих текст и код.

Прямое применение общедоступных моделей через окно чата является достаточно трудозатратным и требует активного участия пользователя и совершения множества повторяющихся действий. В связи с этим, создание апробированного интерактивного инструмента, который бы автоматизировал процесс рецензирования отчетов по лабораторным работам, предоставляя необходимые удобные сервисные возможности для управления проверками, хранения их истории и адаптации под конкретные учебные курсы представляется нам актуальным.

Целью исследования является разработка и апробация программного комплекса для автоматизированного рецензирования отчетов по лабораторным работам студентов с использованием больших языковых моделей, направленного на снижение нагрузки на преподавательский состав и повышение качества, скорости и объективности обратной связи.

Разработанный программный комплекс представляет собой Web-приложение. Бизнес логика реализована с использованием фреймворка NestJS на платформе Node.js, а графический интерфейс в виде одностраничного приложения - с использованием библиотеки React.js. Для хранения информации (курсы, группы, студенты, лабораторные работы, промпты, модели, провайдеры, API-ключи, результаты проверок) используется

встраиваемая СУБД SQLite. Такой стек обеспечивает кроссплатформенность и удобство развертывания как в компьютерной сети, так и на отдельном компьютере.

Программный комплекс не привязан к конкретной LLM, а предоставляет интерфейс для подключения различных LLM через систему провайдеров и API-ключей. Поддерживается работа как с облачными моделями через OpenAI API, так и с моделями, развернутыми локально с помощью платформы Ollama. Для обеспечения надежности взаимодействия с внешними сервисами реализован механизм повторных запросов с настраиваемой задержкой при ошибках.

Процесс проверки включает следующие шаги:

Предварительная настройка: создание дисциплины (дисциплин) и промптов для проверки, создание лабораторных работ и загрузка текстов заданий по ним; выбор и настройка параметров подключения к сервисам, предоставляющим доступ к LLM, задание параметров их работы (размер контекста, температура и др.).

Система будет генерировать запрос к LLM на основе задания к лабораторной работе, текста отчета студента и специального промпта курса, который задает общие критерии оценки. Промпт инструктирует модель вернуть ответ, содержащий оценку, а также текстовый отзыв.

Пример использованного промпта: “Вы преподаватель в университете и эксперт по базам данных и системам управления базами данных, информационным системам и разработке программных приложений для них. Вы выдали студенту задание для выполнение лабораторной работы по учебной дисциплине "Базы данных". Оцени предоставленный студентом отчет по лабораторной работе на: соответствие выполненного задания исходному заданию с учетом варианта студента; правильность и полноту выполнения задания. Выставь оценку отчету студента по шкале от 1 до 10. Оценка ниже 4 выставляется в случае несоответствия варианту или выполнения не всех пунктов задания. Напиши отзыв на выполненную работу и укажи недостатки, если они есть.”

Особенностью системы является возможность настройки текстов промптов под конкретные требования дисциплин и заданий, что позволяет адаптировать анализ к различным типам лабораторных работ.

Предобработка данных. Архив с работами студентов скачивается из системы Moodle, в которой они публиковали свои отчеты, и который имеет определенную структуру. Загруженный архив обрабатывается: отчеты в форматах .docx и .pdf конвертируются в текст с помощью специализированных библиотек; из имен файлов и папок извлекается информация о студентах (ФИО), которая сопоставляется с записями в базе данных.

Выполнение проверки. Реализовано два режима проверки, выбор между которыми осуществляется автоматически на основе паттерна проектирования «Стратегия». Первый – базовый, с использованием одной выбранной LLM. Второй – комплексный, при котором один и тот же отчет анализируется несколькими моделями параллельно. Этот подход позволяет нивелировать возможные слабости отдельных моделей.

При повторной проверке система сравнивает новую версию отчета со старой, определяя, были ли исправлены ранее выявленные недостатки, и формирует новую рецензию и оценку.

Обработка и сохранение результата. Ответ модели анализируется на соответствие ожидаемому формату, валидируется и сохраняется в базе данных. Преподавателю через веб-интерфейс становятся доступны все результаты проверки, сгруппированные по группам и студентам.

Основным результатом исследования является разработанный программный комплекс. Его код и подробное описание приведено в [2]. В качестве используемых для проведения экспериментов LLM были выбраны DeepSeek V3, DeepSeek R1, Llama 3.3

70B, Gemini 2.5 Flash, Qwen QwQ [3]. Все они доступны через API различных провайдеров. Политика ценообразования для запросов с использованием API достаточно гибкая. Ко всем из этих моделей у определенных провайдеров можно получить бесплатный или относительно недорогой доступ, с качеством, приемлемым для решения задач настоящей работы [4].

В ходе апробации проводилась верификация работы программного комплекса на реальных отчетах студентов по дисциплине «Базы данных». Отчеты содержали поясняющие тексты, таблицы, SQL-скрипты и текстовые результаты их выполнения, фрагменты кода на C#. В ходе экспериментов наибольшую эффективность продемонстрировал Gemini 2.5, особенно для анализа объемных отчетов. DeepSeek-R1 также продемонстрировал хорошие результаты при работе с кодом и при анализе текста.

В процессе проверок система успешно выявляла факты несоответствия заданию, синтаксические ошибки в коде и логические недочеты, такие как, например, некорректная нормализацию базы данных, неоптимальные SQL-запросы или способы работы с базой данных с использованием C#. Сформированные моделью списки замечаний и сильных сторон в значительной степени совпадали с теми, которые давал бы преподаватель при внимательном рецензировании. Рецензии Gemini 2.5 отличались исключительной подробностью и глубиной.

Следует отметить особую важность структурирования выданных студентам заданий и содержащиеся в них требования к структуре и форме представления результатов в отчете. В случае недостаточной структурированности, ясности и полноты задания и требований к отчетам LLM могут упускать специфический контекст, заложенный преподавателем в задание. Также LLM не способны напрямую анализировать графический материал, что затрудняет проверку работ с диаграммами и схемами.

Разработанный программный комплекс является мощным и удобным инструментом, который может использоваться для рецензирования работ по различным дисциплинам. При этом решение об окончательной оценке должно оставаться за преподавателем, который использует систему для первичного анализа и формирования развернутой обратной связи. Внедрение данного комплекса в учебный процесс позволит, на наш взгляд, достичь значимых эффектов. Снижается нагрузка на преподавателя за счет автоматизации рутинной части проверки, что высвобождает время для более качественной подготовки к занятиям и индивидуальной работы со студентами. Повышается качество и скорость обратной связи: студенты получают детальный и структурированный разбор своих работ в короткие сроки, что способствует более эффективному освоению материала и своевременному исправлению ошибок. Обеспечивается объективность и стандартизация оценки на этапе первичного разбора работ.

ЛИТЕРАТУРА

1. Scherbakov, D., Hubig, N., Jansari, V., et al. (2025). The emergence of large language models as tools in literature reviews: a large language model-assisted systematic review. Journal of the American Medical Informatics Association, 32(6), 1071-1086.
2. reports-check [Электронный ресурс] / Software Suite for Reviewing Student Laboratory Reports. – Режим доступа: https://github.com/Olgasn/reports-check – Дата доступа: 10.09.2025.
3. Models [Электронный ресурс] / Artificial Analysis. – Режим доступа: https://artificialanalysis.ai/leaderboards/models – Дата доступа: 10.09.2025
4. OpenRouter [Электронный ресурс] / OpenRouter. – Режим доступа: https://openrouter.ai/. – Дата доступа: 10.09.2025.

Automation of Reviewing Laboratory Reports Using Large Language Models

A.D. Asenchik, V.A. Vorobyev
Sukhoi State Technical University of Gomel

In the context of the digital transformation of higher technical education, innovative technologies are being actively introduced into the educational process to increase its efficiency. One of the most time-consuming aspects of teaching IT disciplines remains the review of laboratory reports. The peculiarity of such works lies in their complex structure and the large number of unique task variants. Student reports in programming, databases, or networking technologies contain not only textual descriptions but also fragments of program code, tables, diagrams, SQL queries, and the results of their execution. Checking such works requires the teacher to assess compliance with formal requirements, analyze the logic of the presented solutions, and verify the correctness of the syntax and semantics of the code, which, given a large number of students and task variants, leads to significant time costs and the risk of subjective evaluation.

The emergence and development of large language models (LLMs) open up new opportunities for automating the analysis and evaluation of texts and program code where applicable. Modern LLMs demonstrate high efficiency in content analysis due to their ability to perform deep semantic parsing of complex technical texts, process structured data (code, formulas), and generate meaningful evaluative judgments based on identifying logical relationships in the presented material. An important capability of LLMs is identifying not only obvious errors but also hidden problems, such as violations of design principles in the code or logical inconsistencies in students' explanations, which significantly exceed the capabilities of traditional automated verification systems. LLMs demonstrate a high ability to understand context and identify logical relationships in texts [1], making them particularly suitable for assessing works containing both text and code.

Direct use of public models through a chat window is quite labor-intensive and requires active user participation and many repetitive actions. Therefore, the creation of a tested interactive tool that automates the process of reviewing laboratory reports, providing convenient service capabilities for managing reviews, storing their history, and adapting them to specific courses, seems relevant.

The purpose of the study is to develop and test a software suite for automated reviewing of students' laboratory reports using large language models, aimed at reducing the workload on teaching staff and improving the quality, speed, and objectivity of feedback.

The developed software suite is a web application. The business logic is implemented using the NestJS framework on the Node.js platform, and the graphical interface is a single-page application built with the React.js library. The SQLite embedded DBMS is used to store information (courses, groups, students, lab works, prompts, models, providers, API keys, and review results). This stack ensures cross-platform compatibility and ease of deployment both on a local network and on an individual computer.

The software suite is not tied to a specific LLM but provides an interface for connecting various LLMs via a system of providers and API keys. It supports both cloud-based models accessed through the OpenAI API and locally deployed models using the Ollama platform. To ensure reliable interaction with external services, a retry mechanism with configurable delays for handling errors is implemented.

The review process includes the following steps:

Preliminary setup: creating courses and review prompts, creating lab works and uploading their assignments; selecting and configuring parameters for connecting to services that provide access to LLMs, and setting their operational parameters (context size, temperature, etc.).

IX International Scientific and Methodological Conference "Problems of Modern Education in a Technical University" 2025, Gomel, P.O. Sukhoi Gomel State Technical University, October 23–24, 2025

The system generates a request to the LLM based on the lab assignment, the student's report text, and a special course prompt that defines general evaluation criteria. The prompt instructs the model to return a response containing both an assessment and textual feedback.

Example prompt used: "You are a university professor and an expert in databases, DBMSs, information systems, and software application development. You have given a student a lab assignment for the course 'Databases'. Evaluate the student's lab report for: compliance with the original task and the student's variant, and the correctness and completeness of the work performed. Assign a grade from 1 to 10. A score below 4 should be assigned if the variant does not match or all task points are not completed. Write feedback on the work and point out any shortcomings if present."

A key feature of the system is the ability to customize prompts to meet the specific requirements of courses and assignments, allowing the analysis to be adapted to different types of lab work.

Data preprocessing. Archives with student reports are downloaded from the Moodle system, where students uploaded their reports, and which has a specific structure. The uploaded archive is processed: reports in .docx and .pdf formats are converted into text using specialized libraries; information about students (full names) is extracted from filenames and folders and matched with database records.

Execution of the review. Two review modes are implemented, with selection performed automatically using the Strategy design pattern. The first is the basic mode using a single selected LLM. The second is a complex mode, where the same report is analyzed by multiple models in parallel. This approach helps mitigate potential weaknesses of individual models.

During a repeated review, the system compares the new version of the report with the previous one to determine whether previously identified deficiencies have been corrected and generates a new review and grade.

Processing and saving results. The model's response is analyzed for compliance with the expected format, validated, and saved in the database. Through the web interface, the teacher can access all review results grouped by groups and students.

The main result of the study is the developed software suite. Its code and detailed description are provided in [2]. The LLMs used in the experiments were DeepSeek V3, DeepSeek R1, Llama 3.3 70B, Gemini 2.5 Flash, and Qwen QwQ [3]. All of them are available via various providers' APIs. The pricing policy for API requests is quite flexible. Free or relatively inexpensive access to these models is available from certain providers, with quality sufficient for the purposes of this study [4].

During testing, the software suite was verified on real student reports for the course 'Databases'. The reports contained explanatory texts, tables, SQL scripts and their execution results, and C# code fragments. In the experiments, Gemini 2.5 demonstrated the highest efficiency, especially in analyzing large reports. DeepSeek-R1 also showed strong performance when working with code and text analysis.

During the reviews, the system successfully identified cases of task non-compliance, syntax errors in the code, and logical deficiencies such as incorrect database normalization, suboptimal SQL queries, or inefficient C# database interaction methods. The model-generated lists of remarks and strengths largely coincided with those that a teacher would make during manual review. Gemini 2.5 reviews were particularly detailed and insightful.

It is important to emphasize the significance of structuring assignments and their requirements for report format and structure. When assignments and requirements are insufficiently clear or structured, LLMs may miss specific contextual details implied by the instructor. LLMs also cannot directly analyze graphical materials, making it difficult to assess works containing diagrams or charts.

The developed software suite is a powerful and convenient tool that can be used to review works across various disciplines. However, the final grade should remain at the instructor's discretion, while the system serves for preliminary analysis and detailed feedback generation. The implementation of this system in the educational process can, in our opinion, achieve significant effects. The workload on instructors is reduced through the automation of routine review tasks, freeing time for higher-quality preparation and individual work with students. The quality and speed of feedback increase: students receive detailed and structured reviews promptly, facilitating more effective learning and timely error correction. Objectivity and standardization of assessment are ensured at the initial review stage.

REFERENCES

1. Scherbakov, D., Hubig, N., Jansari, V., et al. (2025). The emergence of large language models as tools in literature reviews: a large language model-assisted systematic review. Journal of the American Medical Informatics Association, 32(6), 1071-1086.

2. reports-check [Online Resource] / Software Suite for Reviewing Student Laboratory Reports. Available at: https://github.com/Olgasn/reports-check (Accessed: 10.09.2025).

3. Models [Online Resource] / Artificial Analysis. Available at: https://artificialanalysis.ai/leaderboards/models (Accessed: 10.09.2025).

4. OpenRouter [Online Resource] / OpenRouter. Available at: https://openrouter.ai/ (Accessed: 10.09.2025).