DOI: 10.15514/ISPRAS-2026-38(5)-18

A Method for Automatic Verification and Correction of Document Formatting in Educational Contexts

V.M. Akkuzina, ORCID: 0009-0006-9119-5430 <viktoria.akkuzina@yandex.ru>
O.L. Vikentyeva, ORCID: 0000-0002-8991-4719 <ovikenteva@hse.ru>
HSE University, 38, Studencheskaya st., Perm, 614070, Russia.

Abstract. This paper proposes a method for the automatic verification and correction of formatting in educational and research documents, designed to support the diverse local formatting standards used by academic institutions. Existing tools are either rigid rule-based systems bound to a single institutional template, incapable of automatic correction, or machine learning approaches that analyze document structure without providing deterministic, rule-based correction. To close this gap, the proposed method combines three components: a tree-structured, extensible template that formalizes local formatting rules for each paragraph type; a classification algorithm that identifies paragraph types from a combination of visual, structural, and contextual features; and a correction mechanism that maps detected inconsistencies to concrete corrective actions. The method was implemented as a client-server prototype built on ASP.NET Core and the OpenXML library, with the formatting template stored as structured XML. The prototype was evaluated on a sample of 10 student reports (1,018 paragraphs) previously reviewed by teaching assistants at HSE University – Perm. Paragraph classification reached an accuracy of 87.2%, with most confusions arising between lists, plain text, and headings in documents lacking style-based formatting. Formatting inconsistencies were detected with a recall of 90.8%, and 83.1% of the identified violations were resolved automatically without introducing new inconsistencies. Automatic verification of a 15–25-page document took about one minute, compared with the 15–30 minutes reported by most teaching assistants for manual review, reducing verification time by more than a factor of ten. These results demonstrate that combining an adaptable rule template with multi-feature paragraph classification enables accurate, transparent, and reusable formatting verification across different institutional standards, while automatic correction remains limited to locally resolvable inconsistencies.

Keywords: automatic formatting verification; formatting correction; paragraph classification; formatting template; educational documents; formatting automation.

For citation: Akkuzina V.M., Vikentyeva O.L. A Method for Automatic Verification and Correction of Document Formatting in Educational Contexts. Trudy ISP RAN/Proc. ISP RAS, 2026, vol. 38, issue 5, pp. 325-336. DOI: 10.15514/ISPRAS-2026-38(5)-18.

Метод автоматической проверки и исправления форматирования текстовых документов в образовательной среде

В.М. Аккузина, ORCID: 0009-0006-9119-5430 <viktoria.akkuzina@yandex.ru>
О.Л. Викентьева, ORCID: 0000-0002-8991-4719 <ovikenteva@hse.ru>
Национальный исследовательский университет «Высшая школа экономики» – Пермь, Россия, 614070, г. Пермь, ул. Студенческая, д.38.

Аннотация. В работе предлагается метод автоматической проверки и исправления форматирования образовательных и научных документов, учитывающий разнообразие локальных стандартов оформления, принятых в отдельных учебных заведениях. Существующие решения либо жёстко привязаны к одному институциональному шаблону и не поддерживают автоматическое исправление, либо основаны на машинном обучении и анализируют структуру документа без детерминированной, основанной на правилах коррекции. Для устранения этого разрыва предложенный метод объединяет три компонента: древовидный расширяемый шаблон, формализующий локальные правила форматирования для каждого типа абзаца; алгоритм классификации, определяющий тип абзаца по совокупности визуальных, структурных и контекстных признаков; и механизм коррекции, сопоставляющий обнаруженные нарушения с конкретными корректирующими действиями. Метод реализован в виде клиент-серверного прототипа на базе ASP.NET Core и библиотеки OpenXML, при этом шаблон форматирования хранится в виде структурированного файла на языке XML. Апробация проводилась на выборке из 10 студенческих отчётов (1018 абзацев), ранее проверенных учебными ассистентами НИУ ВШЭ – Пермь. Точность классификации абзацев составила 87,2%, наибольшее число ошибок связано с разграничением списков, обычного текста и заголовков в документах без стилевого форматирования. Полнота выявления нарушений форматирования составила 90,8%, а 83,1% обнаруженных нарушений были устранены автоматически без внесения новых несоответствий. Автоматическая проверка документа объёмом 15-25 страниц занимала около одной минуты, тогда как большинство учебных ассистентов сообщали о затратах в 15-30 минут на ручную проверку документа сопоставимого объёма, то есть время проверки сокращается более чем в десять раз. Полученные результаты показывают, что сочетание адаптируемого шаблона правил с классификацией абзацев по нескольким признакам обеспечивает точную, прозрачную и переносимую между разными институциональными стандартами проверку форматирования, тогда как автоматическое исправление пока ограничено локально устранимыми нарушениями.

Ключевые слова: автоматическая проверка форматирования; исправление форматирования; классификация абзацев; шаблон форматирования; образовательные документы; автоматизация форматирования.

Для цитирования: Аккузина В.М., Викентьева О.Л. Метод автоматической проверки и исправления форматирования текстовых документов в образовательной среде. Труды ИСП РАН, 2026, том 38, вып. 5, стр. 325–336 (на английском языке). DOI: 10.15514/ISPRAS-2026-38(5)-18.

1. Introduction

Modern educational and research institutions impose strict requirements on the formatting of academic documents, including theses and scientific articles. Failure to comply with these standards may result in lower grades or rejection of publication, making proper formatting an essential part of both educational and research activities. Although national standards such as GOST 7.32–2017 exist, there is no unified set of formatting requirements for academic and scientific texts. Universities and research organizations often apply local regulations that supplement or modify GOST provisions. This diversity of formatting rules complicates the automation of formatting verification.

The problem is aggravated by the scale of verification: the number of student reports is large, while instructors' time is limited. Teaching assistants are often involved to accelerate the process, but their assessments tend to be subjective and inconsistent, which can lead to discrepancies in grading even within the same group. Manual verification also demands sustained concentration and familiarity with local regulations, placing a considerable burden on instructors and assistants and reducing the efficiency of the educational process.

Existing solutions fall into two main categories: rigid rule-based systems tied to specific institutional templates (e.g., Normokontrol, MAPDoc) that lack adaptability and correction capabilities, and machine learning approaches focused on structural analysis without providing deterministic, rule-based correction mechanisms. Thus, current tools are either bound to fixed templates or unable to combine an adaptable representation of formatting rules with automatic correction.

Therefore, the aim of this study is to develop a method for the automatic verification and correction of formatting in textual documents within the educational context. The method includes:

1) formalization of formatting rules in the form of a tree-like template that specifies requirements for different paragraph types;
2) an algorithm for paragraph type classification to ensure accurate application of the template;
3) mechanisms for automatic correction and verification of formatting in accordance with local regulations.

The work is guided by the following research questions:

1) RQ1: How can formatting rules from diverse local standards be formally represented to ensure machine readability and human interpretability?
2) RQ2: What combination of visual, structural, and contextual features is most effective for accurately classifying paragraph types in educational documents?
3) RQ3: To what extent can a hybrid approach – combining paragraph classification with rule-based correction – reduce the time and improve the quality of formatting verification compared to manual methods?

Novelty and Contribution. The scientific novelty lies in a method that integrates a tree-structured, extensible formatting template with a multi-feature paragraph classification algorithm, enabling both accurate detection of inconsistencies and their automatic correction. Unlike prior rule-based tools, the method adapts to the local standards of any educational institution without code changes and provides deterministic correction through low-level document manipulation; unlike machine learning approaches, it requires no training data and yields deterministic, interpretable results.

2. Literature review

Modern word processors, such as Microsoft Word and Google Docs, offer built-in formatting tools; however, their functionality does not include systematic verification of compliance with formatting standards. Moreover, these built-in tools are primarily focused on spelling and grammar at the sentence level, which does not address the need for comprehensive control over document structure and formatting.

Nevertheless, more advanced approaches to automatic text processing are actively developing in the scientific literature. For example, a shift from sentence-level correction to document-level revision has been proposed, utilizing a corpus containing edits made by professional editors to academic texts [1]. Other studies focus on rule-based systems that combine syntactic n-grams and morphological analyzers for automatic grammar correction [2-4]. However, these methods do not address the problem of formatting in educational documents.

A number of studies focus on automating formatting verification and compliance checking in educational settings. One example is the student-developed desktop application "HSEReportChecker" [5], which allows users to configure custom formatting templates and verify DOCX documents. However, the study does not disclose the algorithm used to match formatting rules to specific paragraphs, lacks functionality for automatic correction of detected inconsistencies, and the application itself is not publicly available.

Another example is the "MAPDoc" system, which performs formatting verification by comparing documents with a predefined reference template containing a fixed sequence of styles and structure [6]. Although this approach demonstrates effectiveness on controlled data, it proves insufficiently flexible for real student papers, where document structure may vary and formatting is often inconsistent. Similarly, the "Normokontrol" program, developed at a Kazakh university, effectively checks the formatting of thesis papers, but its functionality is tightly bound to the internal regulations of a specific institution and does not support adaptation to other standards [7].

Beyond these solutions, the problem of automating formatting verification and compliance is addressed in numerous studies that examine the issue from various angles, including the analysis of common formatting errors [8], automatic formatting assessment, document structure analysis and processing, challenges and prospects of automation [9], as well as direct proposals for solving the problem [10-14].

With the advancement of machine learning, new approaches to automatic formatting verification have emerged. Increasingly, machine learning methods are being applied, including paragraph type classification based on a range of features, and the use of transformer architectures to analyze the semantic structure of documents [15-17].

These approaches enable a shift from rigid template matching to probabilistic models capable of accounting for contextual and stylistic variability. However, existing solutions are rarely adapted to educational documents, where it is essential to consider local standards and specific stylistic categories.

Research in the fields of typography and web design [18] demonstrates mature methods of automatic formatting, but these are primarily focused on visual composition and do not incorporate semantic classification of document elements. Attempts to transfer such approaches to academic texts encounter the challenge of structural ambiguity: applying formatting rules requires prior classification of paragraph types, a task typically performed manually or based on predefined markup. This issue is discussed in both survey studies [19] and works proposing specific solutions, such as the use of formatting templates to rigidly define rules for structural elements [20-21].

Another aspect of the problem is the significant variation in formatting approaches: style-based, direct, and hybrid methods are used, which complicates the automatic identification of paragraph types and, consequently, the selection of appropriate formatting rules. This issue is similar to the challenges encountered when analyzing a document's table of contents, which may be generated either automatically or manually. Such structural ambiguities – despite visual similarity – hinder the algorithmic interpretation of document elements. This point is emphasized by I.A. Samoylova [22], who notes that "even when two documents appear visually identical, their internal, programmatic structure may differ significantly," underscoring the need for a formalized description of the internal structure of the text.

Moreover, an analysis of recent publications on electronic document processing [23, 24] highlights the importance of selecting the appropriate target format. Specifically, studies show that the structure and metadata of documents in DOCX, PDF, and ODT formats differ significantly, necessitating the development of specialized algorithms. Accordingly, this study focuses on DOCX documents, which are the most commonly used in educational practice.

3. Materials and methods

3.1 Paragraph type classification

To implement the proposed method for automatic verification and correction of text document formatting in an educational environment, DOCX documents are considered, containing key paragraph types that form the set:

$$P = \left\{ \begin{array}{c} normal,\ heading,\ list,\ table,\ image, \\ image\ caption,\ table\ caption, \\ table\ heading\ row,\ table\ cell \end{array} \right\} \#, \quad (1)$$

where each element corresponds to a specific paragraph type.

For each paragraph $p \in D$ in the document D, it is necessary to determine its type $t \in P$ and verify its compliance with the applicable rule set R_t.The paragraph classification function is defined as:

$$f_{cls}\colon (\ C_p\ , S_p, N_p, O_p \mapsto t), \quad (2)$$

where:

- C_p – textual content of the paragraph;
- S_p – paragraph style (as used in the document);
- N_p – paragraph context (preceding and following elements in the document);
- O_p – paragraph object type (textual, tabular, graphical).

This definition formalizes a combined approach that takes into account:

- Visual features – paragraph formatting parameters such as font, size, and line spacing;
- Structural features – the paragraph's association with a document object (text, table, image) and the presence of special elements (e.g., image captions, header rows);
- Contextual features – the paragraph's position relative to other document elements, such table captions.

An example implementation of such a classifier as a decision tree is presented in Fig. 1. The tree is traversed from the paragraph style downward: if the style identifies a heading, the paragraph is labeled as a heading and its level is derived from the heading hierarchy; otherwise, the object type O_p is examined, and paragraphs associated with a table or an image are further distinguished – by position and context – into table heading rows, cells, or captions and into images or image captions. Plain text and lists are separated by the presence of list markers or numbering and by indentation. The most frequent confusions arise between lists and plain text and between lists and headings, and they worsen when style-based formatting is absent. Idealized conditions are shown, under which captions are required for both tables and images; in practice, the presence of captions is defined by the formatting template and is taken into account during classification.

3.2 Rule formalization and verification

After paragraph classification, each type is assigned a set of formatting rules:

$$\mathrm{R}_t = \{\mathrm{r}_1, \mathrm{r}_2, \dots, \mathrm{r}_\mathrm{n}\}, \quad (3)$$

where each rule r_i describes a paragraph property critical for compliance with the standard (e.g., line spacing). The rules are selected on the basis of formatting standards and of the errors most frequently made by students, such as incorrect indentation and non-standard list markers, table cell margins, and violations of heading hierarchy. They are formalized as a tree-like template (table 1), which ensures extensibility and machine readability.

Verification is reduced to three stages: obtaining the paragraph type t_p using f_{cls}; identifying the set of formatting inconsistencies:

$$\mathrm{E(p)} = \left\{\mathrm{r} \in \mathrm{R}_{\mathrm{t_p}} \,|\, \neg\rho(\mathrm{p},\mathrm{r})\right\}, \quad (4)$$

where $\rho(p, r)$ is a Boolean predicate that is true if paragraph p satisfies rule r; and generating a list of remarks:

$$L = \{(p, E(p))\}, \quad (5)$$

which can be presented either as a report covering all paragraphs or as the original DOCX annotated with comments.

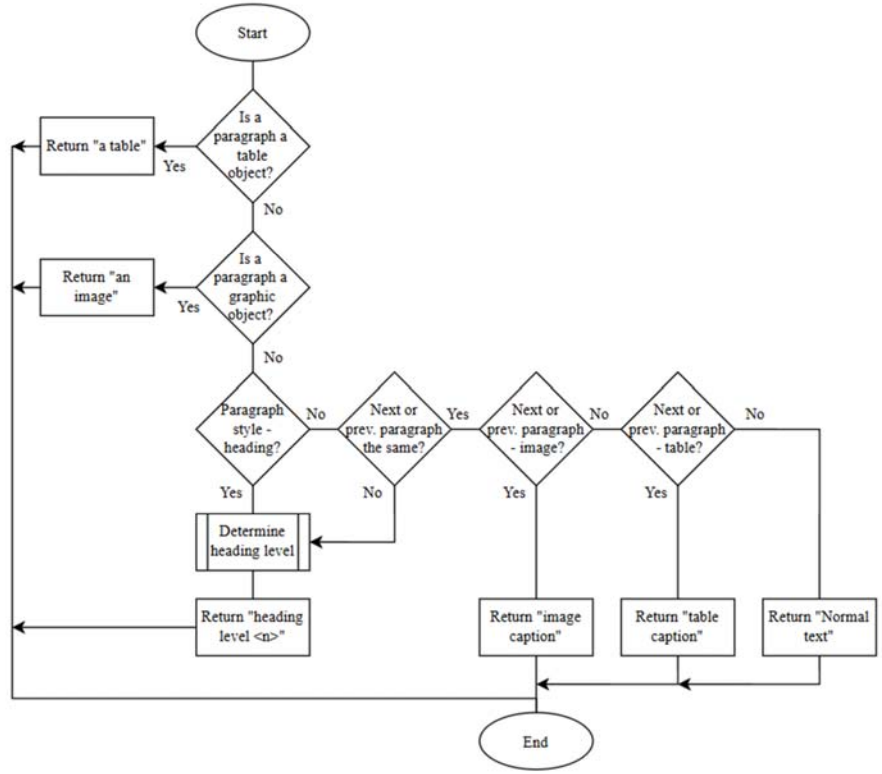

Fig. 1. Paragraph type classification.

Correction maps each remark to a corrective action and applies these actions to the document, yielding a modified document D' in which the detected inconsistencies are resolved. The correction function is:

$$\mathrm{D}' = \omega(\mathrm{D}, \mathrm{R}) = \left\{\mathrm{p_i}' = \mathrm{A}\big(\mathrm{E}(\mathrm{p_i})\big)\right\}_{\mathrm{i}\,=\,1}^{\mathrm{n}}, \quad (6)$$

where ω denotes the transformation of the original document D according to the template rules R, $E(p_i)$ the inconsistencies detected in paragraph p_i, $A(E(p_i))$ the corresponding corrective actions, and p_i' the resulting corrected paragraph.

3.3 Prototype implementation

To validate the proposed method, a prototype was implemented as a client-server web application. The server side is built with ASP.NET Core Web API (.NET 8); DOCX processing relies on the DocumentFormat.OpenXml Library (OpenXML SDK 3.3.0), which is used to extract paragraphs, classify their types, and apply both style-based and direct formatting. Data are stored in PostgreSQL 16 (users, documents, templates, and operation results, accessed through Entity Framework Core 8) and in a MinIO object store (original and corrected documents); the formatting template itself is kept as structured XML. The client side is implemented in HTML, CSS, and JavaScript, served by nginx, and communicates with the server through a RESTful API (JSON over HTTPS, with JWT-based authentication). The system is containerized with Docker Compose (database, backend, and frontend services).

Table 1. Structure of the formatting template (fragment).

Element	Element Type	Description
Template	Formatting Template	Root element containing a set of paragraph type definitions
ParagraphType	Paragraph Type	Defines the type of paragraph
IsCaption	Formatting Property	Indicates whether the given type includes a caption
CaptionLocation	Formatting Property	Specifies the location of the caption relative to the type
LineSpacing	Formatting Property	Specifies the line spacing for the type
BeforeSpacing	Formatting Property	Specifies the spacing before the type
AfterSpacing	Formatting Property	Specifies the spacing after the type

Extensibility is achieved through several design patterns. The Strategy pattern provides a separate correction/verification strategy for each supported element type, so that adding a new type reduces to implementing a new strategy without modifying existing code; the Chain of Responsibility pattern represents the heading hierarchy as a linked list traversed recursively to apply corrections at each level; and the Template Method pattern fixes the invariant steps of the verification and correction algorithms in a base class while leaving type-specific steps abstract. Verification and correction are performed on the server side, while the user interface allows manual adjustment of the automatic paragraph classification before processing.

The system is deployed on Render Cloud as three cooperating web services (Fig. 2). The frontend service serves static files through nginx and communicates with the user's browser over HTTPS; the backend service exposes a REST API (ASP.NET Core 8, port 8080) over HTTP and handles document processing, classification, and correction requests from the frontend. The backend, in turn, connects to a managed PostgreSQL 16 instance over TCP/SQL for relational data, to a MinIO storage service through the S3 API for original and corrected documents, and to an SMTP server (smtp.yandex.ru, port 587, TLS) for delivering processed documents by email. This separation of frontend, backend, database, and object storage into independent services allows each component to be scaled or replaced independently of the others.

4. Results

For the experimental evaluation of the proposed method, a sample of 10 student reports was selected. These reports had previously been reviewed by teaching assistants of the educational program “Information Systems Development for Business” at HSE University – Perm, and were randomly drawn from the archive of student works of previous years. To avoid bias, the same assistant did not evaluate both the original and the system-corrected version of a document. The sample comprised 1,018 paragraphs, excluding empty lines and code fragments, and all documents were manually annotated by paragraph type (plain text, headings, lists, captions for figures and tables, images, and tables).

The effectiveness of the method was assessed along four dimensions:

- Paragraph classification accuracy – the proportion of paragraphs whose type was correctly identified by the system relative to manual annotation;
- Detection recall – the proportion of reference formatting inconsistencies found by the system;
- Correction completeness – the proportion of formatting violations resolved by the system;
- The ratio of time spent on automatic versus manual verification.

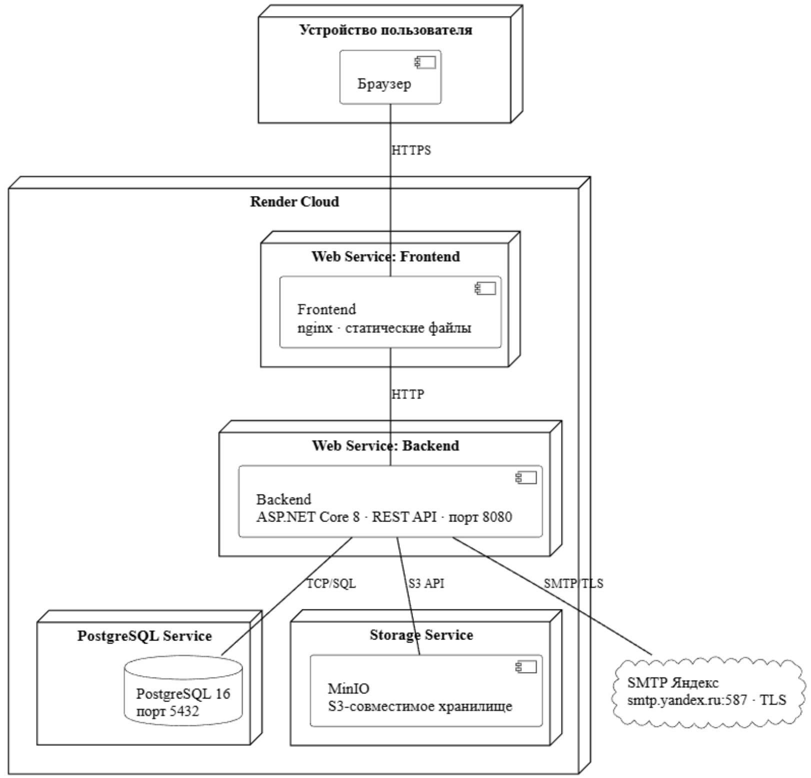

Fig. 2. System deployment diagram.

The detection and correction subsystems were evaluated in isolation from classification: after automatic classification, paragraph types were manually corrected through the system's interface before running detection and correction. This isolation was intentional – it allows each subsystem to be assessed independently of classification errors. A full end-to-end evaluation, in which classification errors (the 87.2% accuracy reported below) propagate to the downstream stages, is left for future work.

At the first stage, the accuracy of the paragraph classification algorithm was evaluated by comparing automatic classification with manual annotation. The method achieved an accuracy of 87.2%. The largest number of errors occurred in distinguishing lists from plain text and lists from headings; classification quality decreased in documents lacking style-based formatting, since the current algorithm identifies headings primarily by style. Accuracy was computed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (7)$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

At the second stage, the detection of formatting inconsistencies was evaluated on documents with manually corrected paragraph types, and the system's remarks were compared with those of the teaching assistants. The detection recall was 90.8%, indicating a strong ability to identify formatting violations. In several cases the system detected issues not noted by the assistants, which highlights its potential for improving formatting quality; such remarks may have been overlooked by reviewers due to their perceived insignificance or human error. Across the documents in the sample, recall varied by 12.4%, indicating that performance remained relatively stable for documents of differing structure and complexity. Recall was computed as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (8)$$

At the third stage, the automatic correction was evaluated: documents with manually corrected paragraph types were processed by the system, and the corrected versions were submitted to independent teaching assistants unfamiliar with the authors' writing styles. The system did not introduce any new formatting inconsistencies; the violations that remained corresponded to original inconsistencies that had not been detected rather than to erroneous corrections. The resulting correction completeness – the share of resolved violations, computed analogously to recall (8) over the set of formatting violations – was 83.1%. The absence of newly introduced inconsistencies indicates that the applied corrections were reliable.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (9)$$

The final stage compared the time cost of verification. On average, the system verified a 15–25-page document in about one minute. According to a survey of ten teaching assistants, 60% reported spending up to 15 minutes on a document of similar length, and another 30% between 15 and 30 minutes.

The effectiveness metrics are summarized in Table 2.

Table 2. Effectiveness metrics of the method.

Metric	Value, %
Paragraph Classification Accuracy	87.2
Formatting Inconsistency Detection Recall	90.8
Recall Variation	12.4
Correction Precision	83.1

Thus, the method reduces formatting-verification time by more than a factor of ten. Over a typical academic semester, for a workload of 90 reports, this amounts to a saving of approximately 12 hours of assistant work.

5. Discussion

5.1 Answering the research questions

5.1.1 Formal representation of rules (RQ1)

The tree-structured XML template (Table 1) provides a machine-readable, human-editable format. It separates rule definition from implementation, allowing adaptation to different local standards without code changes. This satisfies both machine readability and interpretability.

5.1.2 Effective combination of features (RQ2)

The combination of visual (font, spacing), structural (object type, presence of captions), and contextual (position relative to tables/figures) features enabled 87.2% classification accuracy. The analysis of errors revealed that contextual features alone were insufficient when style-based formatting was absent; the hybrid feature set proved effective but highlights the need for more robust features in unstructured documents.

5.1.3 Time and quality improvement (RQ3)

The hybrid approach reduced verification time by more than tenfold, achieved a detection recall of 90.8%, and a correction precision of 83.1% without introducing new inconsistencies. These results confirm a substantial improvement over manual methods in both speed and consistency.

5.2 Comparison with existing approaches

Compared with rule-based tools such as Normokontrol and MAPDoc, which are bound to a fixed institutional template or a predefined sequence of styles, the proposed method expresses rules in a configurable template and can therefore be adapted to different local standards – including GOST-based rule sets – without code changes, while additionally providing automatic correction, which these tools lack. Compared with machine-learning approaches to paragraph classification and document-structure analysis, the method requires no training data and yields deterministic, interpretable decisions and corrections, at the cost of lower robustness on documents without style-based formatting.

5.3 Limitations

First, classification accuracy depends on the presence of style-based formatting; documents formatted manually reduce performance. Second, the rule set is limited to common paragraph types; more complex elements (e.g., multi-level lists, tables of contents) are not yet covered. Third, the experimental sample consisted only of student reports; effectiveness may vary for other document types. Finally, the relatively small data set limits the generalizability of results.

5.4 Future work

Future work will focus on (1) improving classification using transformer-based models (e.g., LayoutLMv3) to handle style-less documents, (2) expanding the rule set to include complex structures, and (3) developing a user-friendly interface for non-technical users to create and modify formatting templates.

6. Conclusion

This paper proposed a method for automatic verification and correction of document formatting tailored to educational contexts. The method combines a tree-structured formatting template with a multi-feature paragraph classifier, enabling both detection of inconsistencies and their automatic correction. Experimental evaluation on student reports demonstrated high accuracy, detection recall, and correction precision, while reducing verification time by an order of magnitude. The method addresses the limitations of existing solutions by providing flexibility, transparency, and integrated correction. Future work will extend the approach to handle more complex document structures and improve classification robustness.

References

[1]. Mita M., Sakaguchi K., Hagiwara M., Mizumoto T., Suzuki J., Inui K. Towards automated document revision: grammatical error correction, fluency edits, and beyond. In Proc. 19th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2024), Mexico City, 2024, pp. 251-265. DOI: 10.48550/arXiv.2205.11484.

[2]. Sidorov G., Gupta A., Tozer M., Catala D., Catena A., Fuentes S., et al. Rule-based system for automatic grammar correction using syntactic n-grams for English language learning (L2). In Proc. 17th Conf. on Computational Natural Language Learning: Shared Task, Sofia, 2013, pp. 96-101. Available at: https://aclanthology.org/W13-3613, accessed 15.10.2025.

[3]. Liu W., Zhao, C. Li Y., Cai C., Liu H., Qiu R., et al. A method for English paragraph grammar correction based on differential fusion of syntactic features. PLOS ONE, 2025, vol. 20, no. 7, e0326081. DOI: 10.1371/journal.pone.0326081.

[4]. Bryant C., Yuan Z., Qorib M.R., Cao H., Ng H.T., Briscoe T. Grammatical error correction: a survey of the state of the art. Computational Linguistics, 2023, vol. 49, no. 3, pp. 643-701. DOI: 10.48550/arXiv.2211.05166.

[5]. Городнов И.Э., Полищук Я.В., Ханжин А.Е. Разработка приложения для автоматизации проверки документации на соответствие требованиям оформления. Сборник материалов III студенческой научно-практической конференции имени Л. Л. Любимова. – Пермь: Редакционно-издательский отдел НИУ ВШЭ – Пермь, 2025, с. 218-223. / Gorodnov I.E., Polishchuk Ya.V., Khanzhin A.E. Development of an application for automating documentation compliance checking. Proc. III Student Scientific-Practical Conf. named after L.L. Lyubimov. Perm, HSE University – Perm Publishing Office, 2025, pp. 218-223 (in Russian). Available at: https://perm.hse.ru/editorial_publishing/Lyubimov_Conference, accessed 15.10.2025.

[6]. Стариченко Б.Е., Устинов М.А. Программа автоматизации контроля оформления текстовых документов // Педагогическое образование в России, 2018, № 8, с. 163-168. DOI: 10.26170/po18-08-22. / Starichenko B.E., Ustinov M.A. Program for automating control of text document formatting. Pedagogical Education in Russia, 2018, no. 8, pp. 163-168 (in Russian). DOI: 10.26170/po18-08-22.

[7]. Самойлова И.А. Некоторые особенности поэтапного алгоритма программы для проверки дипломных работ на нормоконтроль. Молодой ученый, 2016, № 16, с. 51-53. / Samoylova I.A. Some features of a step-by-step algorithm of a program for checking degree theses for compliance with formatting standards. Young Scientist, 2016, no. 16, pp. 51-53 (in Russian). Available at: https://www.elibrary.ru/item.asp?id=26528242, accessed 15.10.2025.

[8]. Шахова И. С. Автоматизация процессов нормоконтроля отчётной документации студентов: концепция программного решения. Информатика и образование, 2024, т. 39, № 2, с. 78-91. DOI: 10.32517/0234-0453-2024-39-2-78-91. / Shakhova I.S. Automation of normative control processes for student reporting documentation: a software concept. Informatics and Education, 2024, vol. 39, no. 2, pp. 78-91 (in Russian). DOI: 10.32517/0234-0453-2024-39-2-78-91.

[9]. Соколов А.М. Нормоконтроль в техническом вузе: проблемы и пути их решения // Высшее образование сегодня, 2015, № 8, с. 26-28. / Sokolov A.M. Norm control in a technical university: problems and ways of their solution. Higher Education Today, 2015, no. 8, pp. 26-28 (in Russian). Available at: https://www.elibrary.ru/item.asp?id=24247753, accessed 15.10.2025.

[10]. Кобец Е.А., Насыров Н.Ф., Комаров М.С., Горлушкина Н.Н. Алгоритм формализации требований к оформлению документов для сервиса автоматизированного нормоконтроля. Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки, 2021, № 7, с. 89-95. DOI: 10.37882/2223-2966.2021.07.15. / Kobets E.A., Nasyrov N.F., Komarov M.S., Gorlushkina N.N. An algorithm for formalizing document formatting requirements for an automated norm control service. Modern Science: Actual Problems of Theory and Practice. Series: Natural and Technical Sciences, 2021, no. 7, pp. 89-95 (in Russian). DOI: 10.37882/2223-2966.2021.07.15.

[11]. Соколов А.А., Дворянкин А.М., Ужва А.Ю. Разработка метода автоматизации процесса нормоконтроля технической документации. Известия ВолгГТУ, 2013, № 22, с. 114-118. / Sokolov A.A., Dvoryankin A.M., Uzhva A.Yu. Development of a method for automating the norm control process of technical documentation. Izvestia VolgGTU, 2013, no. 22, pp. 114-118 (in Russian). Available at: https://www.elibrary.ru/item.asp?id=2089191, accessed 15.10.2025.

[12]. Siddique I., Kanse V. R., Ansari A. Compliance verification process automation. Int. J. for Research in Applied Science and Engineering Technology, 2022. DOI: 10.22214/ijraset.2022.40363.

[13]. Нестеров Г.А., Шестакова А.О., Карякин Ю.Е. Автоматизация проверки научно-технической документации. Математическое и информационное моделирование: материалы Всероссийской конференции молодых учёных (Тюмень, 17–21 мая 2021 г.). Тюмень: Изд-во Тюменского гос. университета, 2021, вып. 19, с. 97-101. / Nesterov G.A., Shestakova A.O., Karyakin Yu.E. Automation of scientific and technical documentation verification. Proc. All-Russian Conf. of Young Scientists "Mathematical and Information Modeling" (Tyumen, May 17–21, 2021). Tyumen, Tyumen State University Publ., 2021, issue 19, pp. 97-101 (in Russian). Available at: https://www.elibrary.ru/item.asp?id=47170634, accessed 15.10.2025.

[14]. Shende A., Mullapudi M., Challa N., Enhancing document verification systems: a review of techniques, challenges, and practical implementations. Int. J. of Computer Engineering & Technology, 2024, no. 15, pp. 16-25. DOI: 10.17605/osf.io/hvq8e.

[15]. Nasyrov N., Komarov M., Tartynskikh P., Gorlushkina N. Automated formatting verification technique of paperwork based on gradient boosting on decision trees. Procedia Computer Science, 2020, vol. 178, pp. 365-374. DOI: 10.1016/j.procs.2020.11.038.

[16]. Berezhkov A., Martsinkevich V. Web tool for automated document formatting verification. In Proc. 33rd FRUCT Conf., Helsinki, 2023, pp. 375-381. Available: https://www.elibrary.ru/item.asp?id=53943947, accessed 15.10.2025.

[17]. Nasyrov N., Fedorov D., Bezgin A., Mazein N., Tikhonova K., Gorlushkina N. Paragraph classification and text analysis for automated document formatting: developing and refining approaches. In Lecture Notes in Networks and Systems, Cham: Springer, 2025. DOI: 10.1007/978-3-031-96177-9_14.

[18]. Gupta G., Niranjan S., Shrivastava A., Sinha R. Document layout analysis and classification and its application in OCR. In Proc. IEEE EDOCW, Hong Kong, 2006, p. 58. DOI: 10.1109/EDOCW.2006.29.

[19]. Hurst N., Li W., Marriott K. Review of automatic document formatting. In Proc. ACM Symposium on Document Engineering (DocEng'09), New York, 2009, pp. 99-108. DOI: 10.1145/1600193.1600217.

[20]. Kumar M., Kalyan K., Aka A., Raya S., Subbarayudu Y., Almusawi M. et al., Enhanced automated word formatter. In AIP Conf. Proc., 2025. DOI: 10.1063/5.0261678.

[21]. Zhigalova M., Sukhov A. System of automated checking of textual document design rules. Information Content and Processing, 2015, no. 2, pp. 164–180. Available at: https://publications.hse.ru/pubs/share/folder/0rdkqezioy/185788493.pdf, accessed 15.10.2025.

[22]. Самойлова И.А. Сложность реализации автоматизированной проверки дипломных работ на соответствие техническим нормам. Молодой учёный, 2016, № 15, с. 149-152. / Samoylova I.A. Complexity of implementing automated verification of degree theses for compliance with technical standards. Young Scientist, 2016, no. 15, pp. 149-152 (in Russian). Available at: https://www.elibrary.ru/item.asp?id=26475412, accessed 15.10.2025.

[23]. Мартсинкевич В.И., Ларионова Г.С., Терещенко В.В., Ситникова К.А., Горлушкина Н.Н. Анализ возможностей парсинга электронных текстовых документов для автоматизации нормоконтроля. Экономика. Право. Инновация, 2022, с. 39-49. DOI: 10.17586/2713-1874-2022-3-39-49. / Martsinkevich V.I., Larionova G.S., Tereshchenko V.V., Sitnikova K.A., Gorlushkina N.N. Analysis of electronic text document parsing capabilities for automating norm control. Economics. Law. Innovation, 2022, pp. 39-49 (in Russian). DOI: 10.17586/2713-1874-2022-3-39-49.

[24]. Mokhnacheva Y.V. Document types indexed in WoS and Scopus: similarities, differences, and their significance in the analysis of publication activity. Scientific and Technical Information Processing, 2023, vol. 50, pp. 40-46. DOI: 10.3103/S0147688223010033.

Информация об авторах / Information about authors

Виктория Михайловна АККУЗИНА – бакалавр по направлению «Программная инженерия» НИУ ВШЭ в Перми. Сфера научных интересов: автоматизация проверки и коррекции форматирования документов, классификация абзацев.

Viktoria Mikhailovna AKKUZINA – graduated her bachelor's degree in software engineering at the HSE in Perm. Her research interests include: automated document formatting verification and correction, paragraph classification.

Ольга Леонидовна ВИКЕНТЬЕВА – доцент кафедры информационных технологий в бизнесе на факультете социально-экономических и компьютерных наук в НИУ ВШЭ в Перми. В сферу научных интересов входят: CASE-технология, Анализ и моделирование бизнес-процессов, объектно-ориентированное программирование, объектно-ориентированное моделирование, проектирование систем, управление проектами, активные методы обучения.

Olga Leonidovna VIKENTYEVA is an associate professor of the Department of Information Technology in Business at the Faculty of Socio-Economic and Computer Sciences at the HSE in Perm. Her research interests include: CASE technology, Analysis and modeling of business processes, object-oriented programming, object-oriented modeling, system design, project management, active learning methods.