

УДК 004.42+004.932+006.83

DOI <https://doi.org/10.36994/2788-5518-2025-02-10-15>

АЛГОРИТМИ АВТОМАТИЗОВАНОЇ ПЕРЕВІРКИ ПАРАМЕТРІВ АКАДЕМІЧНИХ ДОКУМЕНТІВ

Свінцицька О. М., к.е.н., доцент, доцент кафедри комп'ютерних наук, Державний університет «Житомирська політехніка», Житомир, Україна. <https://orcid.org/0000-0002-2613-2437>, svintsytska2018@gmail.com

Коротун О. В., к.п.н., доцент, доцент кафедри комп'ютерних наук, Державний університет «Житомирська політехніка», Житомир, Україна. <https://orcid.org/0000-0003-2240-7891>, korotun-o@ztu.edu.ua

Марчук Д. К., старший викладач кафедри комп'ютерних наук, Державний університет «Житомирська політехніка», Житомир, Україна. <https://orcid.org/0000-0001-8675-8047>, kipz_mdk@ztu.edu.ua

Желізко В. В., асистент кафедри комп'ютерних наук, Державний університет «Житомирська політехніка», Житомир, Україна. <https://orcid.org/0009-0001-4178-6631>, kkn_zhvv@ztu.edu.ua

Анотація. У статті розглядаються теоретичні та практичні аспекти автоматизованої перевірки форматування текстових документів формату docx. Проведений аналіз сучасних літературних джерел дозволив виявити прогалини в існуючих рішеннях, які здебільшого зосереджені на автоматичному створенні документів або перевірці окремих параметрів, і не охоплюють повного циклу структурного та форматного контролю. Дослідження свідчать, що завдяки автоматизації перевірки оформлення можна знизити кількість помилок більш ніж на 40 %, особливо у звітній документації формату docx. У зв'язку з цим, актуальність дослідження зумовлена потребою у створенні уніфікованої системи, здатної автоматично аналізувати структуру документа, перевіряти параметри оформлення та формувати інтегральну оцінку якості.

Метою дослідження є проектування та розробка системи автоматизованої перевірки форматування документів формату docx, що порівнює параметри оформлення з вимогами ДСТУ 3008:2015 та внутрішніми стандартами закладів вищої освіти.

У межах цієї статті авторами запропоновано модульну архітектуру програмного прототипу, яка включає StyleAnalyzer – для аналізу стилів тексту, FormatValidator – для перевірки параметрів сторінки, структури та нумерації, і ReportGenerator – для автоматичного формування звіту. Реалізацію алгоритмів здійснено засобами мови Python із використанням бібліотеки python-docx. Результати експериментального тестування підтвердили ефективність створеного прототипу, який може бути використаний для автоматизації процесів форматування та стандартизації академічних документів. Запропонована архітектура дозволяє уніфіковано перевіряти до 95 % вимог форматування ДСТУ 3008:2015, скорочуючи час контролю одного документа у середньому на 70 % порівняно з ручною перевіркою.

Ключові слова: форматування документів, docx, python-docx, інтегральна оцінка, структурний аналіз, автоматизація оформлення.

ALGORITHMS FOR AUTOMATED VERIFICATION OF ACADEMIC DOCUMENT PARAMETERS

Oleksandra Svintsytska, Candidate of Economic Sciences, Associate Professor, State University “Zhytomyr Polytechnic”, Zhytomyr, Ukraine. <https://orcid.org/0000-0002-2613-2437>, svintsytska2018@gmail.com

Olha Korotun, Candidate of Pedagogical Sciences, Associate Professor, “Zhytomyr Polytechnic”, Zhytomyr, Ukraine. <https://orcid.org/0000-0003-2240-7891>, korotun-o@ztu.edu.ua

Dmytro Marchuk, Senior Lecturer of the Department of Computer Sciences, “Zhytomyr Polytechnic”, Zhytomyr, Ukraine. <https://orcid.org/0000-0001-8675-8047>, kipz_mdk@ztu.edu.ua

Viktor Zhelizko, Assistant of the Department of Computer Science, “Zhytomyr Polytechnic”, Zhytomyr, Ukraine. <https://orcid.org/0009-0001-4178-6631>, kkn_zhvv@ztu.edu.ua

Abstract. The article discusses the theoretical and practical aspects of automated formatting verification of docx text documents. An analysis of current literature has revealed gaps in existing solutions, which are mostly focused on automatic document creation or verification of individual parameters and do not cover the full cycle of structural and formatting control. Research shows that automation of formatting verification

can reduce the number of errors by more than 40 %, especially in docx format reporting documentation. In this regard, the relevance of the study is determined by the need to create a unified system capable of automatically analyzing the structure of a document, verifying formatting parameters, and generating an integrated quality assessment.

The purpose of the study is to design and develop a system for automated verification of docx document formatting that compares formatting parameters with the requirements of DSTU 3008:2015 and internal standards of higher education institutions.

Within the scope of this article, the authors propose a modular architecture for a software prototype that includes StyleAnalyzer for analyzing text styles, FormatValidator for checking page parameters, structure, and numbering, and ReportGenerator for automatically generating reports. The algorithms are implemented using Python with the python-docx library. The results of experimental testing confirmed the effectiveness of the created prototype, which can be used to automate the formatting and standardization of academic documents. The proposed architecture allows for unified verification of up to 95 % of the formatting requirements of DSTU 3008:2015, reducing the time required to check a single document by an average of 70 % compared to manual verification.

Key words: document formatting, docx, python-docx, integral assessment, structural analysis, automation of formatting.

Вступ

Незважаючи на активний розвиток сфери освітніх технологій (EdTech), її частка у загальному обсязі IT-проектів залишається відносно невеликою. За даними рейтингу аналітичної платформи dou.ua, у 2025 році EdTech-проекти складали лише 5 % серед 10-ти найбільш активних напрямів сучасного IT-ринку. При цьому слід зазначити, що представлений рейтинг не охоплює низку стратегічно важливих для освітньої галузі та країни сфер, як-от LegalTech, GovTech, науково-дослідницькі IT-системи або автоматизація документообігу. Отже, рішення в сфері нашого дослідження, зокрема, автоматизації перевірки документів є особливо важливими, оскільки є невід'ємною складовою загальної освітньої технології, спрямованої на підвищення якості підготовки, звітності й наукової комунікації. Розробка алгоритмів перевірки форматування документів стосується напрямів наукових досліджень на перетині EdTech і Data Processing, поєднуючи аналітичні методи з цифровими інструментами контролю якості документації.

В цій статті авторами запропоновано та науково обґрунтовано створення програмного продукту, який дозволить автоматизувати процес перевірки якості оформлення академічних робіт формату docx. Формат DOCX обрано авторами як найбільш прийнятний у порівнянні з PDF, TXT та RTF, оскільки він має відкриту XML-структуру, яка дозволяє програмно ідентифікувати такі елементи документа як заголовки, таблиці, абзаци, стилі, а також здійснювати структурний аналіз та перевірку параметрів для форматування. На основі цього автори статті пропонують власне вирішення проблеми, яке полягає у розробці алгоритмів перевірки форматних параметрів документів формату docx. Дослідження спрямоване на створення прототипу додатку з модулями StyleAnalyzer, FormatValidator та ReportGenerator, які дозволяють: виявляти невідповідності у стилях, відступах, міжрядкових інтервалах та нумерації; порівнювати параметри форматування з установленими стандартами; автоматично формувати звіт про результати перевірки. Параметри форматування документа, такі як шрифт, відступ, міжрядковий інтервал, стиль заголовка подані у вигляді структурованих змінних, що дозволяє перевіряти їх автоматично за допомогою алгоритмів контролю відповідності. Сам алгоритм побудований на основі модульного підходу, що формує послідовну систему обробки даних, від екстракції XML-елементів до порівняння їхніх параметрів зі стандартами, а також пройшов експериментальну перевірку на таких документах як звіти, курсові та статті, що доводить його практичну значимість, підвищує якість академічних робіт і має потенціал до масштабування в освітніх установах.

Отже, дослідження присвячене створенню алгоритмів перевірки форматних параметрів документів формату DOCX, найбільш придатного стандарту для академічного середовища завдяки відкритій XML-структурі, що дає змогу програмно ідентифікувати такі структурні елементи як заголовки, абзаци, таблиці, стилі, та виконувати їх валідацію. Основна частина описує повний цикл проектування, реалізації та експериментальної перевірки алгоритмів перевірки форматування документів формату docx.

Аналіз останніх досліджень та публікацій

Аналіз літератури свідчить, що більшість сучасних інструментів форматування текстових документів працюють не зовсім некоректно в умовах чітко заданих параметрів і не завжди підтримують автоматичну верифікацію відповідності нормативним вимогам. Одні з перших наукових публікацій, які містять певні результати дослідження цієї проблематики, свідчать про зростаючу потребу в уніфікованих алгоритмах, які б поєднували структурний аналіз і форматний контроль текстових документів [3]. На думку авторів Martsinkevich V. et al. [1] та Nasyrov N. et al. [2], автоматизована перевірка дозволяє зменшити кількість помилок оформлення більш ніж на 40 %, особливо у звітній документації. У роботі [2] запропоновано Automated Formatting Verification Technique of Paperwork Elements, що

знижує потребу у ручному контролі, а в статті на тему "Continuous document layout analysis: Human-in-the-loop..." досліджуються сучасні методи безперервного аналізу макета документів (continuous document layout analysis), у яких поєднуються алгоритми машинного навчання та людська участь у циклі контролю (human-in-the-loop) [4].

Про процедуру інструментальної перевірки магістерських кваліфікаційних робіт на академічний плагіат та текстові запозичення з відкритих джерел для встановлення ступеня подібності тексту описано у статті [5]. Вона надає рекомендаційну інформацію, тоді як остаточне рішення про правомірність виявлених запозичень приймає науковий керівник або інші уповноважені особи, враховуючи результати перевірки, специфіку роботи та власні знання у галузі, передбачено відшкодування витрат за повторну перевірку у випадку низького рівня оригінальності (понад 40 %); використання програмного комплексу здійснюється на підставі угоди для забезпечення високої якості підготовки, сприяння академічній доброчесності та підвищення якості освіти через виявлення плагіату.

Метою розробленої системи у [6] для формування та редагування текстових документів визначено автоматизація рутинних процесів при їх створенні, що дозволить скоротити час на написання документів різного виду та зробити їх більш інформативними.

Розробку та вдосконалення методу автоматичної перевірки форматів документів задля забезпечення їх відповідності заданим стандартам представлено у дослідження [7]. Автори проаналізували існуючі підходи, такі як системи на основі правил та машинне навчання, та запропонували модифікований метод, що поєднує структурну та лінгвістичну перевірку. У роботі здійснено порівняльний аналіз нового методу з наявними рішеннями.

Програмну реалізацію методів аналізу та нормоконтролю технічних звітів у форматі .docx, використовуючи технології (.NET, WPF, DocumentFormat.OpenXml) для автоматичної перевірки їх відповідності встановленим вимогам, демонструють дослідники у [8]. Визначене рішення контролює форматування сторінок, послідовність стилів, наявність порожніх рядків та відповідність параметрів сторінок стандартам (A4, Letter). Ключовою особливістю є оптимізація розміру програми, всі виявлені невідповідності фіксуються у звітному файлі. Як зазначають автори, програмне рішення спрощує процес нормоконтролю, підвищує якість та стандартизацію документації (зокрема, пояснювальних записок до кваліфікаційних робіт), зменшуючи витрати часу й ресурсів та сприяючи оптимізації робочих процесів.

У дослідженні [9] описано різні методи, які допоможуть виявити нечіткі дублікати текстових документів за допомогою методів Жаккарда, TF-IDF, Doc2Vec, Universal Sentence Encoder (USE), Bidirectional Encoder Representations from Transformers (BERT). Автори вказують на відмінності кожного з методів, а також описують поетапно комбінований метод виявлення нечітких дублікатів текстових документів, який складається з кроків, а саме: попередня обробка тексту, векторизація, вимірювання подібності, кластеризація, ранжування, застосування моделей машинного навчання. Цей підхід дозволяє зробити процес перевірки більш точним та глибшим.

Запропоновано та реалізовано високоефективний автоматизований підхід для класифікації текстових документів за заданою користувачем темою у роботі дослідників [10]. Він базується на модифікованому статистичному алгоритмі на базі метрики TF-IDF, що вдосконалений за рахунок нормалізації та фільтрації даних, а також багатокритеріального підходу для точнішого розпізнавання відповідності. Зазначене рішення підвищує якість виявлення релевантних документів і може бути застосоване у пошукових системах, рекомендаційних сервісах та інших аналітичних інструментах.

Метою статті є проектування та розробка інтелектуальної системи автоматизованої перевірки форматування текстових документів формату docx, яка забезпечує порівняння параметрів оформлення з нормативними вимогами (ДСТУ 3008:2015) та внутрішніми стандартами закладів вищої освіти.

Викладення основного матеріалу

Для досягнення поставленої мети автори використовували загальні методи теоретичного моделювання, алгоритмічного проектування та експериментальної перевірки програмного прототипу, яким є додаток для перевірки форматування документів формату docx. У межах дослідження застосовано також спеціальні наукові методи, що охоплюють усі етапи розробки – від моделювання до реалізації, зокрема, структурний аналіз – для визначення ієрархії елементів документа та побудови логічної моделі; формальна декомпозиція – для поділу процесу перевірки на окремі підзадачі; метод алгоритмічного проектування – для побудови послідовної системи взаємодії модулів; метод порівняльної валідації – для зіставлення отриманих параметрів форматування із нормативними стандартами. Сукупність цих методів забезпечує наукову обґрунтованість і комплексність розробленої системи перевірки форматування документів docx.

На початковому етапі визначено ключову задачу дослідження – створити інтелектуальну систему, яка дозволить автоматично перевіряти відповідність параметрів оформлення документів стандартам освітніх і наукових установ. Для цього проведено аналіз чинних вимог до форматування академічних документів, таких як шрифти, поля, інтервали, нумерація, стилі заголовків, визначено їх як формальні ознаки, що підлягають алгоритмізації. Зокрема, основним джерелом вимог до такого роду робіт є ДСТУ 3008:2015 «Документація. Звіти у сфері науки і техніки. Структура та правила оформлення», який регламентує загальні принципи побудови та подання текстових документів. Крім того,

кожен заклад освіти має власні методичні рекомендації щодо оформлення курсових, кваліфікаційних та наукових робіт, які деталізують вимоги ДСТУ в частині шрифтів, інтервалів, нумерації сторінок, стилів заголовків і структури розділів. Аналіз цих документів дозволив формалізувати параметри оформлення як набір контрольних змінних, що стали основою для розробки та алгоритмізації модулів FormatValidator і StyleAnalyzer системи перевірки форматування документів.

На основі отриманих вимог було спроектовано алгоритмічну модель перевірки, що реалізує модульний принцип обробки документів та забезпечує повний цикл аналізу форматних параметрів документа, від екстракції стилів до формування підсумкового звіту. Модульний підхід обрано для можливості чіткого розмежування функцій системи, забезпечення масштабованості, гнучкості та можливості запуску оновлень. Кожен модуль виконує свою функцію в межах єдиного алгоритмічного процесу, обмінюючись даними через уніфіковану структуру у форматі json.

Отже, перший модуль (StyleAnalyzer) відповідає за аналіз стилістичних параметрів документа, таких як типи шрифтів, розміри, міжрядкові інтервали, відступи, вирівнювання, стилі заголовків. Його розроблено на основі алгоритму аналізу стилів тексту, що здійснює попередню екстракцію цих даних docx-файлу через бібліотеку *python-docx* і формує таблицю ознак для подальшої перевірки.

Другий модуль (FormatValidator) виконує порівняння отриманих параметрів зі стандартами, визначеними на основі ДСТУ 3008:2015, та внутрішніми нормативними документами закладів освіти. Використовує принцип формальної декомпозиції для виконання незалежної перевірки кожного параметра форматування з подальшою агрегацією результатів у систему показників. В межах цього модуля реалізовано алгоритм перевірки параметрів сторінки, який охоплює валідацію полів, орієнтації, нумерації та параметрів форматування тексту. Цей модуль базується

Третій модуль (ReportGenerator) завершує процес перевірки, автоматично формуючи структурований звіт про виявлені відхилення. Звіт може експортуватися у вигляді json або текстового файлу та містить деталізовану інформацію про кожен параметр, що не відповідає встановленим вимогам.

Таким чином, на етапі проектування алгоритмічної моделі було створено систему, яка поєднує аналітичний, валідаційний і звітний рівні обробки даних, що забезпечує комплексність і прозорість контролю форматування академічних документів.

Наступним кроком дослідження стала розробка програмного прототипу, який об'єднує три вищезазначені модулі в єдину систему обробки документів. На цьому етапі було здійснено перехід від алгоритмічного проектування до програмної реалізації, що передбачало вибір інструментів, мови програмування та способів представлення даних. Для реалізації було обрано мову програмування Python, що забезпечує ефективну роботу з XML-структурами формату docx за допомогою бібліотеки *python-docx*, що надає доступ до стилів, абзаців, таблиць і властивостей форматування.

Ключовим принципом побудови прототипу є формалізація параметрів оформлення, коли кожен атрибут документа, наприклад, шрифт, інтервал, відступ, стиль заголовка тощо представлено у вигляді структурованої змінної. Це дозволяє здійснювати автоматичну верифікацію шляхом порівняння значень параметрів із нормативними стандартами за допомогою умовних операторів і правил відповідності. Розроблена програма охоплює повний цикл перевірки документа, що включає:

- екстракцію структурних елементів із файлу формату DOCX;
- аналіз параметрів форматування;
- порівняння їх зі стандартами та нормативними документами;
- генерацію підсумкового звіту.

Такий підхід забезпечує логічну прозорість процесу перевірки, гнучкість оновлень і можливість масштабування системи.

На етапі тестування програмного прототипу використано текстові документи різних типів – звіти, курсові та наукові статті обсягом від 10 до 30 сторінок, які відрізнялися структурою, кількістю розділів і складністю форматування. Перевірка проводилася поетапно, відповідно до логіки роботи модулів системи. Експериментальні результати підтвердили працездатність створеного прототипу та ефективність модульної архітектури, яка забезпечує стабільну й точну перевірку параметрів оформлення документів.

Отже, після опису основних етапів дослідження в контексті поставленої мети, подальша частина роботи присвячена детальному опису алгоритмів перевірки оформлення документів, що реалізують функціональність кожного з розроблених модулів і забезпечують повну автоматизовану перевірку відповідності документів вимогам ДСТУ 3008:2015 та внутрішніх стандартів освітніх установ.

Алгоритм перевірки параметрів сторінки. Метою алгоритму є валідація відповідності параметрів сторінки та нумерації в електронному документі (DOCX) встановленим стандартам оформлення, зокрема, вимогам, що висуваються до курсових, дипломних робіт або інших академічних документів.

На першому етапі виконується перевірка встановлених розмірів полів сторінки для кожної секції $S_i \in P_Sections$. Умова:

$$\forall S_i \in P_Sections: (S_i.LeftMargin = Target_Left) \wedge (S_i.RightMargin = Target_Right) \wedge (S_i.TopMargin = Target_Top) \wedge (S_i.BottomMargin = Target_Bottom)$$

Діаграму послідовності, яка формально описує перший етап алгоритму перевірки розміру полів представлено на рисунку 1.

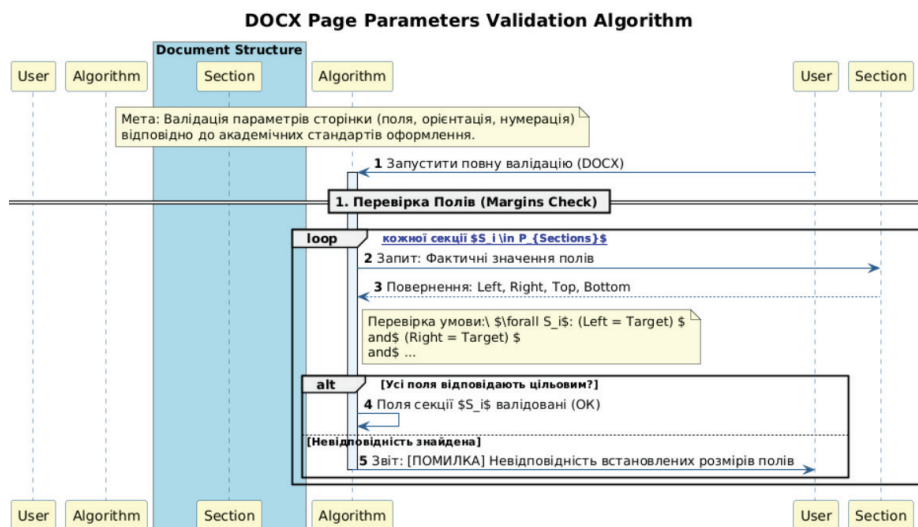


Рис.1. Діаграма послідовності перевірки полів

На другому етапі перевірка, чи не змінена орієнтація (альбомна) там, де вона не дозволена (крім додатків або великих таблиць).

$$\forall S_i \in P_Sections: (S_i.Orientation = Portrait) \vee (S_i.IsException = True)$$

Винятки ($S_i.IsException = True$):

- Секції, що містять Додатки.
- Секції, що містять великі таблиці/діаграми.

Діаграму послідовності, яка формально описує другий етап представлено на рисунку 2.

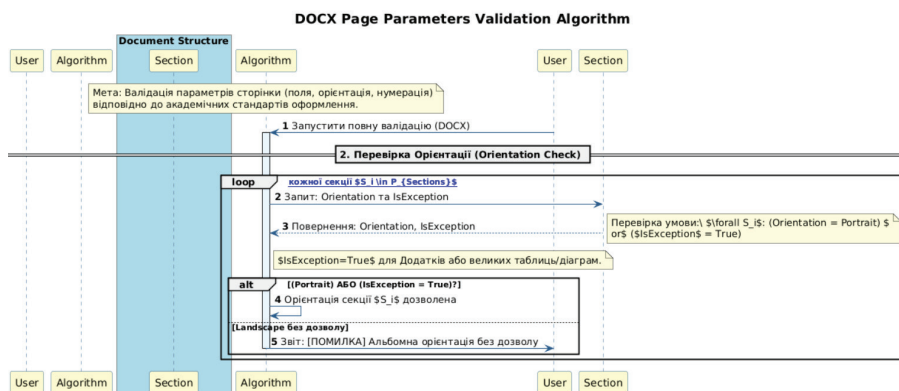


Рис.2. Діаграма послідовності перевірки орієнтації сторінок

Перевірка нумерації наступний крок, валідація наявності та коректного формату нумерації сторінок (наприклад, відсутність нумерації на титульному аркуші, початок нумерації з «Вступу»). Умови: Титульний аркуш – нумерація сторінок повинна бути прихована або відсутня ($PageNumTitlePage = False$).

Початок нумерації – фактичний відлік номерів сторінок має починатися з секції, що містить розділ «ВСТУП» ($PageNumStart = Section("ВСТУП")$).

Нумерація повинна відповідати встановленому формату (наприклад, арабські цифри, розташування знизу по центру).

Діаграму послідовності, яка формально описує алгоритм перевірки параметрів сторінки представлено на рисунку 1.

Алгоритм аналізу стилів тексту. Activity Diagrama чітко показує послідовність кроків та точки прийняття рішень для кожного типу елемента (заголовок, список, основний текст) (рис.3).

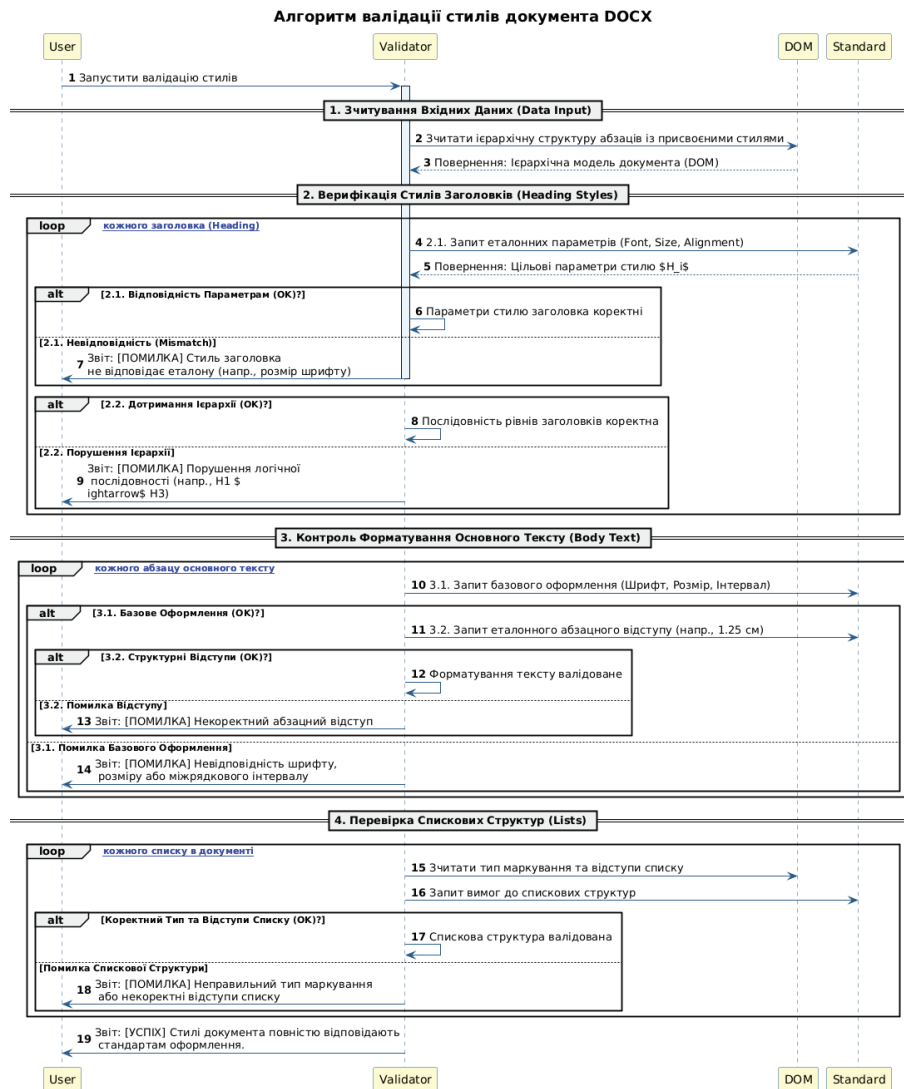


Рис. 3. Алгоритм перевірки стилів тексту

Алгоритм пошуку та валідації посилань. Цей алгоритм забезпечує комплексну перехресну валідацію всіх посилань (зовнішніх джерел, ілюстрацій та таблиць), згаданих у текстовому масиві, шляхом виконання трьох основних, потенційно паралельних, процесів. Алгоритм складається з трьох основних паралельних процесів: пошук посилань на джерела (у квадратних дужках), пошук посилань на ілюстрації/таблиці, і фінальна перехресна перевірка їхньої валідності

1. Пошук посилань у тексті. Використовуються регулярні вирази для пошуку патернів, що відповідають посиланню на джерело у квадратних дужках (наприклад, [1; 5], [5, с. 23]).

2. Валідація посилань на ілюстрації та таблиці. Пошук фраз на зразок «Рис. 1.2» або «Табл. 3» у тексті.

3. Перехресна перевірка (Валідація):

- Джерела: перевірка, чи існує посилання [N] у списку літератури, де N – номер джерела.
- Ілюстрації: перевірка, чи дійсно існує об'єкт з підписом «Рис. 1.2» після першого посилання на нього.

Алгоритм перевірки бібліографії. Даний алгоритм охоплює повний цикл перевірки бібліографії, починаючи від атомізації масиву тексту до структурної та послідовної перевірки кожного елемента.

1. Сегментація записів. Розділення бібліографії на окремі записи (наприклад, за нумерацією або окремими абзацами).

2. Валідація формату (Стандарт). Для кожного запису застосовується набір регулярних виразів для перевірки структури, що відповідає обраному стандарту (наприклад, ДСТУ чи APA).

- Перевірка наявності автора, назви, видавництва/журналу, року, коректне використання розділових знаків (крапки, тире, скісні риски).

3. Перевірка послідовності, чи розташовані записи у правильному порядку (наприклад, алфавітному або порядку згадування).

Після опису основних алгоритмів представимо модель формування результатів перевірки. В основі її побудови покладено механізм структурованого збереження, аналізу і візуалізації виявлених відхилень з метою сформулювати звіт у форматі, придатному як для автоматичної обробки системою, так і для подальшого аналізу користувачем. Модель реалізована за принципами побудови структури даних, логіки фіксації помилок і розрахунку інтегрального показника відповідності оформлення нормативним вимогам.

Результати перевірки зберігаються у структурі, що дозволяє легко генерувати звіт та локалізувати помилку.

```
json
{
  "total_errors": 15,
  "errors_by_category": {
    "page_layout": [
      {"code": "P-001", "message": "Ліве поле сторінки 3 встановлено 25 мм, очікується 30 мм.", "location": "Сторінка 3"},
      //...
    ],
    "text_styles": [
      {"code": "T-005", "message": "Заголовок 'ВСТУП' використано зі стилем 'Normal'. Очікується 'Heading 1'.", "location": "Рядок 120"},
      //...
    ],
    "references": [
      {"code": "R-002", "message": "Знайдено посилання [25], яке відсутнє у списку літератури.", "location": "Сторінка 15"}
    ]
  }
}
```

Алгоритм розрахунку показника оцінки оформлення використовується для надання єдиної, кількісної оцінки якості оформлення. Кожній категорії або конкретному типу виявленої помилки присвоюється коефіцієнт ваги (W_i), що відображає її критичність та вплив на загальну якість документа (Табл.1).

Таблиця 1
Схема визначення відхилень (вага)

Категорія помилки	Тип критичності	Приклад відхилення	Вага (W_i)
Критична	Структурні / Життєво важливі	Некоректні поля, відсутність списку літератури, порушення структури заголовків.	≥ 3.0
Основна (Major)	Вимоги до змісту елементів	Неправильний формат цитування, некоректний стиль заголовка розділу, відсутність обов'язкових елементів бібліографії.	1.5–2.9
Незначна (Minor)	Стилістичні/Естетичні	Неправильний інтервал між абзацами, використання некоректного шрифту для виносок, зайві пробіли.	1.0–1.4

Визначення ваги помилок розраховується як сума (S) добутків кількості виявлених помилок (n_i) кожного типу на відповідний коефіцієнт ваги (W_i).

$$S = \sum_{i=1}^k (n_i \cdot W_i), \quad (1)$$

де k – загальна кількість унікальних типів помилок; n_i – кількість виявлень помилки i -го типу; W_i – коефіцієнт ваги для помилки i -го типу.

Для отримання уніфікованої оцінки у відсотках потрібно нормалізувати отриману суму відносно максимально можливого або встановленого базового штрафу. Максимальний базовий штраф ($S_{\max}$) – це ключовий параметр, який встановлюється як еталон. Інтегральний показник (E) – це відсоток відповідності вимогам. Він розраховується шляхом віднімання частки нормалізованого штрафу від 100 % відповідності ідеальному оформленню.

$$E = 100 \cdot \left(1 - \frac{S}{S_{\max}} \right) \quad (2)$$

Наведемо приклад розрахунку у таблиці 2. При умові, що $S_{max} = 30.0$

Таблиця 2
Приклад розрахунку сумарного штрафу

Помилка (i)	Опис	Вага (Wi)	Кількість	Штраф
1 (P-001)	Порушення лівого поля	3.0 (Критична)	1	3.0
2 (R-002)	Посилання відсутнє у списку	2.5 (Основна)	2	5.0
3 (T-005)	Неправильний шрифт абзаців	1.0 (Незначна)	5	5.0
4 (P-002)	Неправильний міжрядковий інтервал	1.2 (Незначна)	10	12.0
Разом	–	–	–	25.0

Розрахунок інтегрального показника $E = 100 \cdot (1 - 25.0/30.0) = 100 \cdot 16.67\%$. У даному прикладі E становить 16.67 %, це дозволяє об'єктивно оцінити роботу, де 100 % означає повну відповідність вимогам, а нижчий відсоток – необхідність виправлень.

Наведений вище алгоритм було реалізовано у межах експериментального тестування прототипу під час перевірки текстових документів (звіти, курсові, наукові статті). Система автоматично обчислювала показник E для кожного файлу, що дало змогу порівняти якість оформлення різних типів документів. Отримані результати показали, що документи, створені відповідно до методичних рекомендацій закладу освіти, мали інтегральний показник у межах 95–98 %, а документи з порушеннями стилів, відступів і нумерації – 75–90 %, що підтвердило ефективність алгоритму та коректність такого підходу до оцінювання форматування. Таким чином, розроблена модель не лише формалізує процес перевірки, а й забезпечує можливість кількісного порівняння якості оформлення документів на основі об'єктивних показників.

Генерація звіту про перевірку оформлення документів формату docx є завершальним етапом запропонованої нами системи. Саме інтегральна оцінка, яка обчислюється на основі вагових коефіцієнтів помилок, слугує основою для формування змісту звіту: вона визначає загальний рівень відповідності документа нормативним вимогам, а також дозволяє класифікувати порушення за критичністю. Модуль ReportGenerator використовує результати цієї оцінки для автоматичного створення структурованого звіту, що містить деталізовану інформацію про кожне виявлене відхилення, його місцезнаходження та рекомендації щодо усунення. Такий підхід забезпечує прозорість, об'єктивність і зручність оцінювання академічної документації.

Висновки та перспективи подальших досліджень

Для досягнення поставленої мети авторами було спроектовано та реалізовано програмний прототип – інтелектуальну систему перевірки форматування текстових документів формату docx, що поєднує структурний аналіз, алгоритмічну валідацію параметрів оформлення та генерацію інтегральної оцінки якості. Розроблена система охоплює повний цикл обробки документа, від екстракції стилів до формування підсумкового звіту, має тримодульну архітектуру (StyleAnalyzer, FormatValidator, ReportGenerator), яка реалізує функціональність аналізу, перевірки та звітування на основі параметрів форматування згідно ДСТУ 3008:2015 та внутрішніх стандартів ЗВО. В статті наведені відповідні алгоритми та результати експериментального тестування на академічних документах ЗВО, яке підтвердило ефективність запропонованого рішення, його стабільність та точність. Отримані інтегральні показники (від 75 % до 98 %) свідчать про здатність системи об'єктивно оцінювати якість оформлення та зменшувати кількість помилок у академічній документації.

Перспективи подальших досліджень передбачають розширення функціоналу створеного прототипу в частині семантичного аналізу змісту документа, інтеграції з репозитаріями освітніх та наукових робіт, а також створення комплексної платформи оцінки якості, структури та змісту документів у межах EdTech-рішень.

Запропонована система є науково обґрунтованим і технічно реалізованим інструментом, що має потенціал до широкого застосування в академічному середовищі, сприяє підвищенню якості освітнього процесу та автоматизації документообігу.

Література

1. Martsinkevich V., Veselova E. Web Tool for Automated Document Formatting Verification. *Proceedings of FRUCT Conference*. 2020. [Online] URL: <https://fruct.org/publications/volume-33/acm33/files/Mar.pdf>
2. Nasyrov N. Automated Formatting Verification Technique of Paperwork Elements. *Procedia Computer Science*. Elsevier, 2020. Vol. 169. pp. 719–724. DOI: 10.1016/j.procs.2020.02.189. URL: <https://www.sciencedirect.com/science/article/pii/S1877050920324170>
3. Zhigalova M., Morozova A., Bessarabov K. System of Automated Checking of Textual Document Design. *HSE University Publications*. 2015. URL: <https://publications.hse.ru/pubs/share/folder/0rdkqezioy/185788493.pdf>
4. Peña A. Continuous document layout analysis: Human-in-the-loop..., Information Sciences. Elsevier. 2024. [Online]. URL: <https://www.sciencedirect.com/science/article/pii/S1566253524001763>
5. Сидоренко В. В., Кулішов В. С., Гірка С. В. Порядок проведення інструментальної перевірки на академічний плагіат, наявність текстових запозичень у рукописах кваліфікаційних робіт здобувачів вищої освіти освітнього ступеня «магістр» у 2025 році. Біла Церква : БІНПО, 2025. 18 с. URL: <https://lib.iitta.gov.ua/id/eprint/744928>

6. Музичук Д. Р., Войтко В. В., Черноволик Г. О. Розробка засобів системи автоматизованого формування і редагування текстових документів. *Інформаційні технології і автоматизація – 2024* : матеріали XVII міжнародної науково-практичної конференції (Одеса, 31 жовтня – 1 листопада 2024 р.). Одеса : Видавництво ОНТУ, 2024. С. 479–482.
7. Вербовий Д. С., Саяпіна І. О. Метод для автоматичної перевірки документів на відповідність нормативним вимогам. *Системні технології*. 2025. № 3 (158). С. 181–187.
8. Сидорко Д. С., Глухов В. С. Програмна реалізація методів аналізу та нормоконтролю текстових технічних звітів. *Computer systems and networks*. 2024. № 6 (2). С. 208–221. DOI: 10.23939/csn2024.02.208.
9. Заболотня Т., Сечин І. Комбінований метод виявлення нечітких дублікатів текстових документів. *Progressive Science and Achievements* : proceedings of the 4th International Scientific and Practical Conference (May 16–18, 2024). Doha, Qatar, 2024. № 200. С. 229–232.
10. Данильченко В. М. Багатокритеріальне розпізнавання відповідності текстів темі на основі алгоритму TF-IDF. *Зв'язок*. 2025. № 3. С. 36–41. DOI: 10.31673/2412-9070.2025.027728.

References

1. Martsinkevich V., Veselova E. (2020). Web Tool for Automated Document Formatting Verification. Proceedings of FRUCT Conference. [Online]. URL: <https://fruct.org/publications/volume-33/acm33/files/Mar.pdf>
2. Nasyrov N. (2020). Automated Formatting Verification Technique of Paperwork Elements. *Procedia Computer Science. Elsevier*. Vol. 169. Pp. 719–724. DOI: 10.1016/j.procs.2020.02.189. URL: <https://www.sciencedirect.com/science/article/pii/S1877050920324170>.
3. Zhigalova M., Morozova A., Bessarabov K. (2015). System of Automated Checking of Textual Document Design. HSE University Publications. URL: <https://publications.hse.ru/pubs/share/folder/Ordkqezioy/185788493.pdf>
4. Peña A. (2024). Continuous document layout analysis: Human-in-the-loop..., *Information Sciences. Elsevier*. [Online] URL: <https://www.sciencedirect.com/science/article/pii/S1566253524001763> [in Ukrainian].
5. Sydorenko V. V., Kulishov V. S., Hirka S. V. (2025). Poriadok provedennia instrumentalnoi perevirky na akademichnyi plahiat, naiavnist tekstovykh zapozychen u rukopysakh kvalifikatsiinykh robot zdobuvachiv vyshchoi osvity osvithnoho stupenia "mahistr" u 2025 rotsi. *Bila Tserkva : BINPO*. 18 p. URL: <https://lib.iitta.gov.ua/id/eprint/744928> [in Ukrainian].
6. Muzychuk D. R., Voitko V. V., Chernovolik H. O. (2024). Rozrobka zasobiv systemy avtomatyzovanoho formuvannia i redahuvannia tekstovykh dokumentiv. *Information technology and automation – 2024* : materialy XVII mizhnarodnoi naukovo-praktychnoi konferentsii (Odesa, 31 zhovtnia – 1 lystopada 2024 r.). Odesa : Vydavnytstvo ONTU. Pp. 479–482 [in Ukrainian].
7. Verbovyi D. S., Saiapina I. O. (2025). Metod dlia avtomatychnoi perevirky dokumentiv na vidpovidnist normatyvnyvm vymoham. *System technologies*. № 3 (158). Pp. 181–187. [in Ukrainian]
8. Sydorko D. S., Hlukhov V. S. (2024). Prohramna realizatsiia metodiv analizu ta normokontroliu tekstovykh tekhnichnykh zvitiv. *Computer systems and networks*. № 6 (2). Pp. 208–221. DOI: 10.23939/csn2024.02.208 [in Ukrainian].
9. Zabolotnia T., Siechyn I. (2024). Kombinovanyi metod vyjavlennia nechitkykh dublikativ tekstovykh dokumentiv. *Progressive Science and Achievements* : Proceedings of the 4th International Scientific and Practical Conference (May 16–18, 2024). Doha, Qatar. № 200. Pp. 229–232 [in Ukrainian].
10. Danylchenko V. M. (2025). Bahatokryterialne rozpiznavannia vidpovidnosti tekstiv temi na osnovi alhorytmu TF-IDF. *Communication*. № 3. Pp. 36–41. DOI: 10.31673/2412-9070.2025.027728 [in Ukrainian].

Дата першого надходження статті до видання: 27.10.2025

Дата прийняття статті до друку після рецензування: 25.11.2025

Дата публікації (оприлюднення) статті: 26.12.2025