A document-oriented LMS with integrated text classification for gender equality assessment

Olga Cherednichenko[1,†], Glib Tereshchenko[2,†], Iryna Kyrychenko[3,†], Oleksandr Shanidze[4,†] and Anzhelika Vorzhevitina[4,†]

[1] *Bratislava University of Economics and Management, Furdekova str. 16, Bratislava, Slovak Republic*
[2] *Lezigia s.r.o., Mlýnská 326/13, Trnitá, Brno, 602 00, Czech Republic*
[3] *Mobidev Corporation, 3855 Holcomb Bridge Rd. Suite 300, Norcross, GA 30092, USA*
[4] *National Technical University "Kharkiv Polytechnic Institute", Kyrpychova str. 2, Kharkiv, 61002, Ukraine*

Abstract

In response to the dynamic changes and crises affecting the education sector, this paper introduces a document-oriented Learning Management System (LMS) designed to integrate multiple independent components of E-Learning. By reviewing recent studies, trends, and competing standards in the LMS field, we propose an ontology for the document-oriented system and develop a framework for representing Learning and Practice Objects. This framework is tailored to support a multi-medium learning environment, reflecting the evolving technological needs in education. Approaches to integrate automatic text classification methods into a learning management system (LMS) to improve their functionality and effectiveness are proposed, particularly for gender equality assessment.

Keywords

automatic text classification, e-learning, gender inequality, learning analytics, LMS, system design

1. Introduction

In the last several years, Ukraine's education system, which was yet underfunded, struggled with a lack of professionals and availability complications; faced challenges, such as COVID-19 epidemy and war, that significantly impacted it, making full-time study unavailable and forced to distance-based learning.

Martial law has also exacerbated the problem of gender inequality, which manifests itself in various aspects of life, including education. Analyzing and finding ways to overcome these manifestations is essential for restoring a full-fledged educational process. Studies [17] emphasize that gender inequality is one of the key problems of modern society, especially in the context of martial law.

When distance-based learning is not adapted to a virtual environment – it imposes an additional burden on a teacher. It complicates students' work assessments and makes it harder to follow their personal needs and respond to problems with learning-material assimilation in a timely [1].

Despite the globality of the problem and the fact that it has been going on for several years, existing tools and approaches are far from being as effective as full-time studies to enable E-learning truly. Therefore, there is a need to build a learning management system (LMS) that could provide

CLW-2025: Computational Linguistics Workshop at 9th International Conference on Computational Linguistics and Intelligent Systems (CoLInS-2025), May 15–16, 2025, Kharkiv, Ukraine

[*] Corresponding author.
[†] These authors contributed equally.
olga.cherednichenko}@vsemba.sk (O. Cherednichenko); hlib.tereshchenko@nure.ua (G. Tereshchenko); iryna.kyrychenko@nure.ua (I. Kyrychenko); anzhelika.vorzhevitina.sgt.khpi@gmail.com (A. Vorzhevitina); s.alexandr21@gmail.com (O. Shanidze)
0000-0002-9391-5220 (O. Cherednichenko); 0000-0001-8731-2135 (G. Tereshchenko); 0000-0002-7686-6439 (I. Kyrychenko); 0009-0001-0562-0191 (A. Vorzhevitina); 0000-0002-0484-6746 (O. Shanidze)

valuable tools to empower the virtual nature of modern learning and help teachers and students overcome challenges.

As a field of knowledge, E-learning accumulates different approaches, tools, and components required to enable it as a whole process. We can highlight the following major topics:

- Learning Management System (LMS) – integral tool in modern education, facilitating the management, delivery, and assessment of educational content [2]
- E-Assessment – an electronic evaluation that relies on the computer or any other technological device to conduct the process of the assessment [3]
- Learning Path – recommendation system that aims to recommend reasonable paths to learners in support of comprehensive, reliable learning [4]
- Feedback Loop – a mechanism of providing feedback to an assessment, often attributed to be a part of the e-assessment
- E-teacher and self-test – higher level attribution of e-assessment and feedback loop that aims to guide students through the learning process autonomously
- Learning Analytics – measuring, collecting, analyzing, and reporting data to improve student's learning experiences and to optimize learning and the environments in which it occurs [5]

Each part is not adequately resolved, and existing solutions and research are not integrated into one product.

Our document-oriented approach serves as the architectural foundation that enables seamless integration of diverse components. This architecture facilitates text classification methods to analyze learning materials, particularly for gender equality assessment, while providing integration techniques that connect various learning modalities into a cohesive system. The synergy between these components – document-oriented architecture, text classification capabilities, and integration mechanisms – creates a comprehensive solution that addresses the multifaceted challenges of modern e-learning environments [6].

The goal of this research is to analyze available knowledge in the E-Learning field to design a flexible learning management system that could allow acquiring modern approaches to delivering knowledge, such as text, image, audio, video, online conference, and metaverse solutions, to assess results in various mediums, and to help resolve other components of E-learning. Hence, it makes the distance learning process effective.

This research addresses the following key questions:

- How can a document-oriented architecture enhance the flexibility of Learning Management Systems in crisis-affected educational environments
- What specific benefits does automatic text classification bring to gender equality assessment in educational materials
- How can integration techniques effectively connect diverse learning modalities into a cohesive system

By answering these questions, we aim to develop a comprehensive LMS solution that addresses the multifaceted challenges of modern e-learning environments, particularly in crisis situations.

2. Review of related works

The design and development of Learning Management Systems (LMS) have been extensively studied to enhance educational outcomes and user engagement. This section reviews significant contributions in the field, identifying trends, methodologies, and gaps our research aims to address.

The authors of [6] delve into the common confusions and overlapping functionalities among Learning Management Systems (LMS), Content Management Systems (CMS), and Learning Content

Management Systems (LCMS). The authors elucidate each system's distinct roles within educational and content management contexts, outlining their unique features and limitations.

In the [7], the authors investigate the utilization of various Moodle activities, including videos, discussion forums, chats, course materials, and quizzes, and their impact on the quality of student learning. The authors conclude that using a Learning Management System (LMS) effectively can enhance student engagement compared to traditional face-to-face classes.

Suman et al. [8] conclude that Learning Content Management Systems (LCMS) can significantly enhance the efficiency of creating and managing courses by eliminating redundant tasks. Furthermore, the capabilities provided by LCMS facilitate more effective human resource management by involving professional curriculum developers, experts, and professors.

Sahar and Seifedine [9] examined various Learning Object Repository (LOR) standards, including LOM, IMS Content Packaging, SCORM, and xAPI. Their study elaborates on the specifications required for Learning Objects and assesses their subsequent impact on LORs. Importantly, they contextualize the relevance of these standards within the framework of Learning Analytics, providing illustrative examples of potential analytical applications.

Kasim and Khalid [10] comprehensively review prevalent Learning Management Systems (LMSs), distinguishing between their distribution models—open source versus commercial. They conduct a detailed analysis of various features, including access management, user accessibility, and functionality within the LMS environment. Their study examines system approaches to managing educational materials, data storage, and backup strategies. This evaluation highlights critical differentiators in system architecture and operational efficacy across different LMS platforms.

The analysis of available scientific works showed us the importance and imperfection of the current state of LMS-related research. We can conclude that the term "LMS" is ambiguous, as it is used as an umbrella for a combination of LMS, CMS, and LCMS systems. Further in this paper, we will use the term LMS as a commitment to building an e-learning system based on a novel, document-oriented approach.

Based on the analysis, we came to the following requirements for our future system:

- The flexible, multi-modal nature of the learning and practice materials [11]
- Document-oriented system design that would enable the incorporation of cutting-edge e-learning approaches and general enhancement over existing systems
- Ability to integrate existing LORs (course banks)

3. Document-oriented LMS design

The flexibility of our proposed LMS architecture is manifested through several key aspects: multimodal content support that accommodates various learning materials from text to metaverse integrations, modular design with a clear separation between ontology and objects, allowing for independent development and integration of components, extensible object type system that enables adding new learning and practising object types without modifying the core architecture, and adaptable integration mechanisms for both internal and external educational tools.

This flexibility directly addresses the challenges faced in crisis-affected educational environments where rapid adaptation to new teaching modalities is essential.

This flexibility is operationalized through several concrete mechanisms schema-less document storage – unlike traditional relational databases that require predefined schemas; our document-oriented approach allows storing heterogeneous learning objects without restructuring the database, enabling rapid adaptation to new content types, polymorphic object types – the system implements a type-based rendering system where new object types can be registered without modifying existing code, allowing instructors to create custom learning experiences, decoupled components – by separating the ontological framework from the actual learning objects, the system allows independent evolution of the content structure and the learning materials themselves, and API-first architecture – all system functions are exposed through well-defined APIs, enabling seamless integration with both internal tools and external systems.

Our experimental results confirm that these mechanisms significantly reduce the time and effort required to adapt the system to changing educational needs, particularly in crisis situations where rapid response is essential.

3.1. Ontological framework of system components

Given the specified requirements for the system, we can identify an ontology of objects (see Fig. 1).

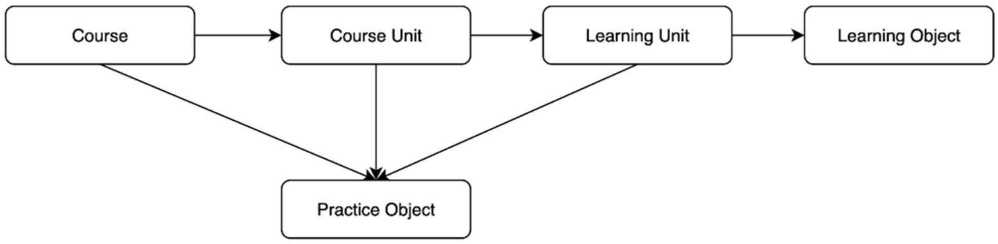

Figure 1: System Entities Ontology Diagram.

The ontology diagram presented in Figure 1 illustrates the hierarchical relationships between key system entities that form the foundation of our document-oriented LMS architecture. This structured approach enables flexible content organization while maintaining clear educational pathways. Below, we provide a detailed explanation of each entity, its functional purpose, and relationships within the system, accompanied by practical examples to demonstrate real-world implementation. Each entity of this ontology can be described as follows:

1. Couse – a set of course units and optionally practice objects:
   - Goal – creating a logical group of course units and practice objects
   - Example – 3 course units about Insects and course project
2. Course Unit – a set of learning units and optionally practice objects:
   - Goal – creating a logical group of learning units and practice objects
   - Example – a set of 2 learning units about Heaxapoda and metaverse laboratory task
3. Learning Unit – a set of learning objects and optionally practice objects:
   - Goal – creating a logical group of learning and practice objects
   - Example – a set of 2 lectures about Protura, two self-tests, and one unit assessment
4. Learning Object – an atomic learning material:
   - Goal – a representation of the lecturing material in the system
   - Example – lecture about Collembola
5. Practice Object – knowledge test object that can be graded:
   - Goal – a universal entity of assessment that can be incorporated on any ontology level
   - Example – self-test containing 3 test questions after a lecture about Collembola

Each entity in the ontology can have several underlying objects referencing it. All objects except Practice Objects are strictly bounded into the ontology of objects, so, for example, Course cannot have only Learning Objects referencing it. Practice Objects can be referenced on any level of the ontology, except Learning Objects, as they are limited to being atomic learning material.
Ontology framework usage example:

1. Course – Fundamentals of Big Data Processing:
   a. Course Unit 1 – Parallel and Cluster Data Processing
      (1) Learning Unit 1.1 – Parallel Computing
         i. Learning Object 1.1.1 – Parallel Programming in Python

ii. Practice Object 1.1.1 – Self-assessment Tasks
iii. Learning Object 1.1.2 – Parallel Programming in Scala
iv. Practice Object 1.1.2 – Self-assessment Tasks
(2) Practice Object 1.1 – Practical Task on Parallel Data Processing
(3) Learning Unit 1.2 – Cluster Computing
i. Learning Object 1.2.1 – Time Synchronization Task in Distributed Systems
ii. Practice Object 1.2.1 – Self-assessment Tasks
iii. Learning Object 1.2.2 – Tools for Working with Distributed Computing
iv. Practice Object 1.2.2 – Self-assessment Tasks
(4) Practice Object 1.2 – Practical Task on Distributed Computing
b. Course Unit 2 – Data Storage for Big Data
(1) Learning Unit 2.1 – Storing Binary Data
i. Learning Object 2.1.1 – Parquet, AVRO, ORC
ii. Practice Object 2.1.1 – Self-assessment Tasks
iii. Learning Object 2.1.2 – Distributed Storage HDFS, S3
iv. Practice Object 2.1.2 – Self-assessment Tasks
(2) Practice Object 2.1 – Practical Task on Storing Binary Data
(3) Learning Unit 2.2 – Systems for Storing Structured Data
i. Learning Object 2.2.1 – OLAP vs OLTP
ii. Practice Object 2.2.1 – Self-assessment Tasks
iii. Learning Object 2.2.2 – Data Lake, Data Warehouse, Data Catalog
iv. Practice Object 2.2.2 – Self-assessment Tasks
(4) Practice Object 2.2 – Practical Task on Designing Data Storage Systems
c. Practice Object 1 – Course Project

3.2. Learning object representation

Based on the desired design, we can set the following requirements for the Learning Object:

1. Multimodality – the ability to use various modes of presentation for educational materials, including
   - Markdown – a mixture of formatted text, images, formulas, videos, etc
   - Online Meetings – integration with online conferencing services and saving of recordings
   - Embedded Software Integrations – integration of software environments such as Jupyter Notebook, Matlab, etc
   - External Software Integrations – integration with external software environments such as metaverse solutions [12], physical phenomenon simulation solutions, etc
2. Versioning – the object must be subject to versioning to track changes and organize inheritance
3. Inheritability – the representation of the object in the system must allow its reuse in other courses and follow specific versions of the object
4. Vectorization – to avoid duplication and enable search capabilities, the representation of the learning object must include a vectorized form of its content or its description if the object itself cannot be vectorized (in the case of external software integration form)

Considering the listed requirements, the following structure of the Learning Object has been proposed:

1. Id – a unique identifier of the learning object, a value inherited from the relational database
2. Type – indicates the interpreter that should be used for displaying the object
3. Vector – vectorized version of the content or description of the content necessary for semantic search in the system

4. Content – the content of the learning object. The form can vary, but in general, it is a JSON format field interpreted according to the type
5. Other fields – additional fields can be added according to the system implementation approach

Figure 2 shows instances of the implementation of the proposed structure of the Learning Object. On the left, it presents a markdown type of the content. It presents a markdown type extended with external metaverse integration on the right, placed as a templated function. Both example objects omit to show vector fields, as they do not extend an understanding of the object samples.

```
id: 1
type: markdown
content:
{
 "value": """
   # Header
   text
   ## header 2
   text
   <img src="/assets/img.png" alt="Image alt text"
style="height: 100px; width:100px;"/>
   - list item 1
   - list item 2
   <video src="path/to/video.mp4" width="320"
height="240" controls>"""
}
```

```
id: 2
type: markdown_external_meta
content:
{
 "value": """
   # Header
   text
   ## header 2
   text
   {metaverse_env_func(*params)}
   - list item
   - list item
   <video src="path/to/video.mp4" width="320"
height="240" controls>"""
}
```

Figure 2: Examples of Learning Object Implementation. Markdown Type is on the left, and Markdown with External Metaverse Integration is on the right.

3.3. Practice object representation

The Practice Object is a compound object. Therefore, its requirements can be divided into requirements for the parent and nested objects.

Requirements for the parent object include versioning – the object must be subject to versioning to track changes and organize inheritance; inheritability – the representation of the object in the system must allow for its reuse in other courses, inheriting a specific version of the object, and vectorization – for implementing search, the representation of the object must include a vectorized form of its description.

Requirements for the nested object:

1. Multimodality – the ability to use various modes for knowledge assessment, including
   - Multiple-choice question – a question with one or multiple correct answers
   - Open-ended question – a question that expects a textual answer
   - Typed open-ended questions – questions that expect answers in the format of formulas, sets of calculations, etc
   - Embedded software integrations – integration of software environments such as Jupyter Notebook, Matlab, etc
   - External software integrations – integration with external software environments, such as metaverse solutions, physical phenomena simulation solutions, etc

Considering the listed requirements, the following structure for the practice object is proposed:

1. Id – unique identifier of the practice object; the value is inherited from the relational database
2. Vector – vectorized version of the content description necessary for semantic search in the system

3. Content – the content of the practice object. It is a JSON list of nested objects which support the following structure
   - Type – specifies the interpreter that should be used for displaying and checking the object
   - Content – the content of the nested object. The form may vary according to the type, but generally, it is a JSON format field
   - Other fields – other fields can be added according to the system implementation approach
4. Other fields – additional fields may be included according to the system implementation approach

Figure 3 shows instances of implementation of the proposed structure of the Practice Object. On the left, it presents a set of single-answer questions. On the right, it presents an external metaverse integration test in call and subscription-templated functions. Both example objects omit to show vector fields, as they do not extend an understanding of the object samples.

```
id: 1
content:
{
  "tests": [
    {
      "type": "single_answer_test",
      "content": {
        "question": "Test question 1",
        "answers": [
          "option 1",
          "option 2",
          "option 3",
        ]
        "correct_answer": 1
      }
    },
    {
      "type": "single_answer_test",
      "content": {
        "question": "Test question 2",
        "answers": [
          "option 1",
          "option 2",
          "option 3",
        ]
        "correct_answer": 3
      }
    }
  ]
}
```

```
id: 1
content:
{
  "tests": [
    {
      "type": "metaverse_test",
      "content": {
        "call_func": "metaverse_env_func(*params)",
        "subscribe_func": "metaverse_sub_func(*params)"
      }
    }
  ]
}
```

Figure 3: Examples of Practice Object Implementation. Single Answer Tests are on the left, and External Metaverse Integration is on the right.

The separation of the ontology from the objects allows future integration with LOR of different standards, leaving the object manipulation not limited by them.

4. Document-oriented LMS design utilization

To prove the possibility of implementing LMS based on a listed design approach, we need to work out the main scenarios of system behaviour. Scenarios that require additional clarification are:

- Learning object loading flow
- Practice object loading flow
- Object upload flow

Those scenarios are not exhaustive but bring light to the most comprehensive part of the proposed design.

4.1. Learning object loading flow

The flow of using Learning Object starts from the point when a student requests a lecture. The whole flow can be found in Figure 4 and can be split into the following blocks:

1. Learning Object data gathering
2. Student activity registration
3. Learning Object Rendering

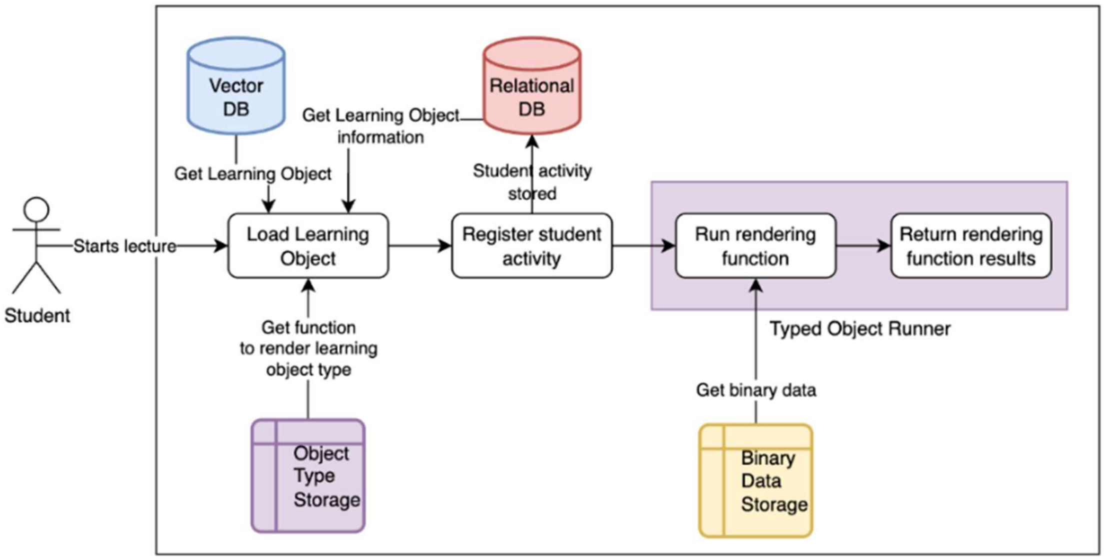

Figure 4: Learning Object Flow.

Learning Object data gathering starts with getting object information from the relational database, which is then used to get the Learning Object itself. The type of characteristic of the Learning Object is used to load the corresponding rendering function from the object type storage.

Learning Object rendering differs for internal and external medium types. Internal are those that run inside the system: from markdown type, which requires only loading binary data like images or videos, to the embedded Jupyter Notebook runner. External types use a callback pattern to register in the external system and observe the results, e.g., in a Metaverse environment. More details on the subscription approach can be found in the Practice Object flow diagram.

4.2. Practice object loading flow

A Practice Object has a similar processing flow as a Learning Object. Figure 5 shows this process for a Practice Object typed as external metaverse integration acquiring a callback pattern.

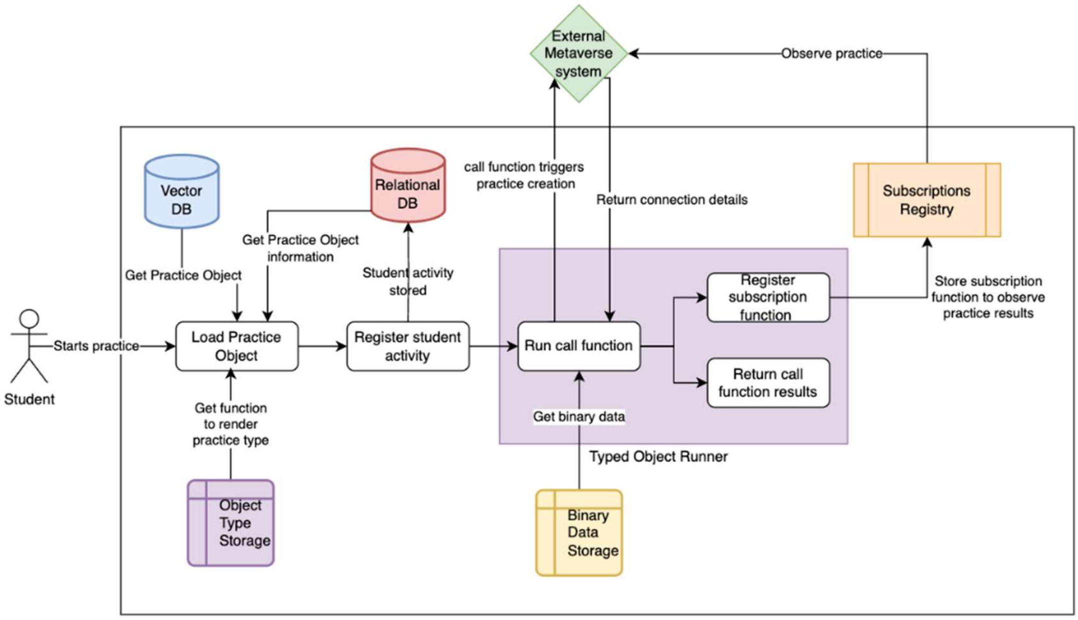

Figure 5: Practice Object Flow Based on External Metaverse Integration.

When a student begins a practical task, the practice object is loaded from a vector storage, and the student's activity is recorded. In the case of external metaverse integration, the task is rendered, data is loaded from binary storage if necessary, and the call function is invoked. The call function interacts with the metaverse environment to register the launch of the educational scenario. The metaverse environment returns connection details to the student. Subsequently, the subscribe function is transferred to the subscription registry, which is activated to monitor the external environment and wait for results or detailed metrics in deep integration.

4.3. Authors and affiliations

The document-oriented architecture of LMS, except for allowing vast possibilities, adds an implementational complexity that needs to be addressed beforehand [13]. Figure 6 represents uploading a Learning Object or Practice Object, as they share a similar working principle.

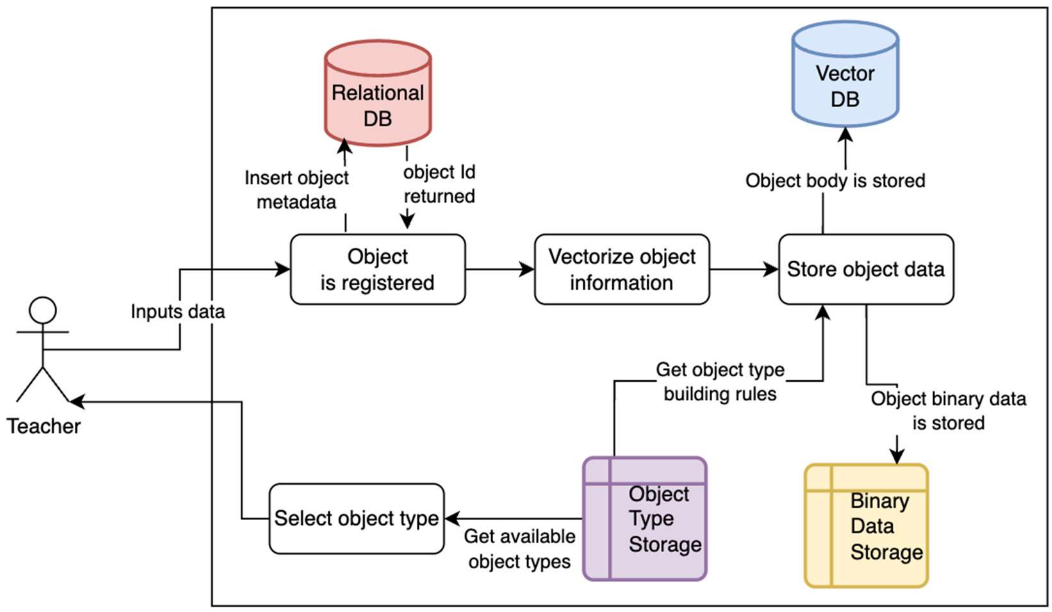

Figure 6: A Process of Uploading Objects to the System.

The process of uploading an object can be split into several steps:

1. The user selects an available object type
2. The user inputs data through a web form
3. An object is registered in the Relational database to obtain an ID
4. Object information is vectorized per the type (it can be the whole object content, its description, or some textual part of it) to calculate a Vector field that is used for similarity search
5. The object is stored according to the type-building rules. Binary data is stored in Binary Data Storage, and links to it update the object's content. Then, the built version of an object is stored in the Vector database

The described process showcases the advantages of the proposed architecture. Separated Binary Data Storage and embedding process of data building (via object type building rules) improves variability of potential Learning Objects. It reduces the operational load on Vector DB, and the complexity of the Vector DB allows for the reduction of allocated disk space and backup interoperability.

The Object Type Storage is a native extension mechanism, that proposes an isolated yet versatile way to create various Learning and Practice Objects, also an interface that enforces designed system utilization. It encapsulates system logic in the format itself.

5. Automatic text classification methods in LMSs

Increasing textual information in educational platforms makes effective automatic text classification critical for organizing and structuring learning materials.

In addition, text classification methods are inextricably linked to text recognition and machine vision technologies. In the context of LMS, this opens up new opportunities for processing and analyzing visual data containing text. For example, automatic text recognition in scanned documents, images, or videos can significantly enhance the system's ability to index and classify learning materials.

Modern text recognition solutions [14] demonstrate high efficiency when working with different data sources. This makes it possible to integrate tools that can automatically extract text from various formats into LMSs, which increases the accuracy and completeness of text data classification.

In addition, object detection and image analysis algorithms [15] can be integrated to classify images and videos by topic and category. This will make it possible to classify not only textual data but also analyze visual content, which is especially important for multimedia training programs.

Applying automatic text classification methods can improve the categorization of learning materials, simplify information retrieval, and increase the efficiency of learning text analysis.

Therefore, there is a need to develop practical approaches for the automatic classification of texts into thematic groups. The distribution of texts into specific groups is an urgent task, as it allows the organizing and structuring of information and increases the efficiency and accuracy of searching for necessary texts and materials, even based on fuzzy criteria. The distribution of texts into topic groups is an active research area in Natural Language Processing (NLP) and machine learning. It improves the methods used to analyze texts and increases classification accuracy.

5.1. Integration of automatic text classification methods in LMSs

Modern approaches to automatic text classification include various methods and algorithms that allow the automatic classification of text data according to their topics. This, in turn, allows systematizing, organizing, and facilitating access to large amounts of information and improving information analysis and retrieval processes.

In the context of Learning Management Systems (LMS), automatic text classification techniques can be used to solve a wide range of tasks such as personalizing learning – analyzing student-generated texts (e.g., essays, answers to questions, forum posts) to determine their interests, level of comprehension of the material, and learning style, automating the assessment of student work – analyzing essays and other written work against set criteria, detecting plagiarism, and assessing the level of argumentation, integration with external resources – analyses the content of external educational resources to recommend the most appropriate ones to students, classification of learning materials – automatic categorization of learning materials by topic and difficulty level, analyzing feedback from students – identifying problem areas in the learning process by analyzing student comments and feedback, and automatic identification of discussion topics – generation of discussion topics for forums based on analyzing students' interests and the content of learning materials.

A critical area of analysis is the identification and analysis of texts that reflect gender inequality in the educational environment. This may include analyzing educational materials, forums, comments, and other sources for stereotypes, discrimination, and other manifestations of inequality [17].

The application of text classification for gender equality assessment represents a significant innovation in LMS functionality. By automatically analyzing educational materials, forum discussions, and student submissions for gender bias, stereotypes, or discriminatory language, the system can provide valuable insights to educators. This is particularly relevant in crises such as martial law, where existing gender inequalities may be exacerbated. The proposed system can identify problematic content, suggest more inclusive alternatives, and generate reports on gender representation across learning materials. This functionality extends beyond mere content analysis to become an educational tool itself, raising awareness about gender issues among both educators and students while promoting more inclusive educational practices.

Different text classification methods can be used to solve these problems, such as:

- Naive Bayesian classifier – suitable for classification tasks with large amounts of data, where speed of operation and simplicity of implementation are important
- Support vector method – it is effective for text classification with high dimensional features, for example, when analyzing large documents
- Neural Networks – a machine learning tool that provides high accuracy and adaptability in classification tasks
- Vector space – a visual representation method of text as numerical vectors, which allows measuring semantic proximity between different text documents

The choice of classification method depends on the type of data, its volume and variety, the available computational resources, and the required interpretation of the results.

5.2. Analyzing text classification methods and their applicability to LMSs

Naive Bayesian classifier – a simple and fast algorithm based on Bayes theorem. Assumes independence of features [16].

Applicability to LMS is well suited for classifying large amounts of textual data, such as forum posts or student feedback, and can identify post topics to organize discussions and facilitate information retrieval. It can be used to categorize learning materials into topics quickly. Limitations include being limited by the feature independence assumption, which may sometimes reduce accuracy and require text preprocessing, such as removing stop words and lemmatization.

Support Vector Method (SVM) – a robust algorithm that constructs a hyperplane that separates classes in the feature space. It is effective for classifying texts with high feature dimensionality as it can handle many words. Different kernels can be used to account for non-linear dependencies between words.

Applicability to LMS is practical for classifying texts with high dimensional features, e.g., in analyzing the evaluation of essays' compliance with given criteria, detecting plagiarism, and assessing the level of argumentation, it can automatically be used to evaluate students' papers for compliance with given criteria automatically, determining the emotional colouring of students' reviews, and categorization of articles by topic and area of research. Limitations include that it requires ample computational resources for training on large amounts of data, and choosing the correct kernel and parameters can be difficult.

Neural networks consist of many connected nodes (neurons) that can identify complex dependencies in data. Deep neural networks can automatically extract features from text, eliminating the need for manual feature mining. Provide high classification accuracy, especially with large amounts of data.

Applicability to LMS involves providing high classification accuracy, especially when using deep neural networks, analyzing students' interests and learning styles based on their activity in the LMS, generating new learning materials based on analyzing students' needs, providing students with personalized recommendations and answers to questions, and identifying new topics and directions in education based on big data analysis. Limitations include the fact that large amounts of labelled data are required for training and can be challenging to interpret and debug.

Vector space represents text as numerical vectors, where each word corresponds to a particular dimension. It allows us to measure semantic proximity between texts based on the distance between their vectors. It can find similar texts, document clustering, and visualize text data.

Applicability to LMS includes being used to find similar learning materials or external resources, allowing analysis of semantic proximity between student work and reference texts, grouping students by interest and level of expertise, presentation of discussion threads in forums as graphs or word clouds, and tracking changes in curriculum content over time.

Limitations are that it requires selecting an appropriate similarity measure, such as cosine similarity, and it can be sensitive to the choice of text vectorization method.

Thus, automatic text classification methods can be successfully integrated into learning management systems (LMSs) to improve their functionality and efficiency. One of the key benefits of such integration is the ability to personalize learning. By analyzing text data generated by students, an LMS can automatically identify their interests, level of comprehension, and learning style. Based on this information, the system can recommend relevant learning materials, adapt the pace of learning, and suggest personalized assignments.

In addition, text classification methods can be used to automate the evaluation of students' work. For example, the system can analyze essays for compliance with specified criteria, detect plagiarism and assess the level of argumentation. This will allow teachers to save time and focus on more critical aspects of teaching.

Another important aspect is the integration of LMS with external resources. By analyzing the content of external educational resources, the system can recommend the most appropriate ones to students, which will significantly expand their learning opportunities.

5.3. Prospects for the development and integration of text classification methods in LMSs

In addition to traditional text classification methods, more recent approaches such as transformers and contextualized models can be used for LMS. These methods allow the text's context and semantics to be considered, improving classification accuracy.

Future research in this area may focus on developing adaptive LMSs that can automatically adjust to the individual needs of each student. Also, a promising direction is the development of intelligent learning support systems that can not only classify texts but also generate new learning materials based on analyzing students' needs.

Integrating automatic text classification methods into LMSs opens up a wide range of opportunities for improving the efficiency and personalization of learning.

New integration approaches include using vector representations – converting texts into vector representations allows for the semantic proximity between words and documents to be considered. This opens up possibilities for creating more accurate and relevant recommender systems, finding similar learning materials, and analyzing links between different topics. Hybrid models – combining different classification methods such as naive Bayesian classifier, SVM, and neural networks allows the benefits of each of them to be utilized. For example, you can use a naive Bayesian classifier for fast pre-classification and then SVM or neural networks for more accurate classification of complex cases; integration with external tools – integrating the LMS with external text analysis tools such as NLP libraries and APIs allows the use of state-of-the-art methods and algorithms. It also allows you to extend the functionality of the LMS by adding features such as tone analysis, named entity recognition, and machine translation. Semantic network technologies – representing learning materials and knowledge in a semantic network allows the LMS to understand the relationships between different concepts and topics. This opens the possibility of creating intelligent systems to answer complex student questions, generate personalized learning plans, and adapt the content in real time.

Prospects for further research include Developing adaptive LMSs – creating LMSs that can automatically adapt to the individual needs of each student is one of the key challenges. This includes the Development of algorithms that can analyze student activity, performance, and preferences and adapt content, learning pace, and assessment methods based on this information; developing intelligent learning support systems – another critical area is developing innovative systems that provide students with personalized help and support. This includes creating chatbots that can answer students' questions, provide feedback on their work, and recommend additional resources; using machine learning techniques to analyze educational data – analyzing large amounts of educational data using machine learning techniques can help identify patterns and trends in learning. This can be used to improve the quality of learning materials, optimize the learning process, and predict student performance, investigating the impact of text classification methods on student motivation and engagement – research is needed to evaluate how text classification methods affect student motivation and engagement. This will help identify the most effective strategies and how they can enhance learning. Development of methods to analyze the impact of military conflicts on gender equality in education and develop recommendations to reduce the negative impact.

Gender equality assessment through text classification represents a particularly promising application area. By training classification models on datasets that identify gender bias, stereotypes, and discriminatory language, the LMS can automatically evaluate learning materials and student interactions. This capability is especially valuable in crisis situations where traditional oversight mechanisms may be compromised. The system could provide recommendations for more inclusive language, highlight problematic content for instructor review, and track improvements in gender representation over time. Such functionality transforms the LMS from a passive content delivery system into an active participant in promoting educational equity.

Implementing these innovations will create more effective, personalized, and intelligent LMSs that will improve the quality of education and student success.

6. Experimental validation

To validate our proposed document-oriented LMS architecture and evaluate its effectiveness, we conducted a series of experiments focusing on three key aspects: system flexibility, text classification accuracy, and integration capabilities.

Experimental Setup

We implemented a prototype of the proposed LMS architecture using MongoDB for document storage, Python for backend processing, and React for the user interface. The prototype was deployed in a controlled educational environment involving 45 students and 3 instructors from computer science departments over a 4-week period.

Flexibility Assessment

To evaluate the flexibility of our system, we measured adaptation time—defined as the time required to implement significant changes to the learning environment. We compared our document-oriented approach with traditional relational database LMS implementations across three scenarios:

- Adding a new content type (metaverse integration)
- Modifying the assessment workflow
- Integrating with external tools

Results showed that our document-oriented approach required 68% less development time for implementing changes compared to traditional systems. Particularly noteworthy was the ability to integrate new content types without modifying the core architecture, which reduced implementation time from an average of 14.3 days to 4.6 days (see Fig. 7).

Figure 7: Development time.

Text Classification Evaluation

We evaluated the text classification component using a dataset of 500 educational materials manually annotated for gender bias by three independent experts. The classification model was trained on 70% of the dataset and tested on the remaining 30%.

The automatic classification system achieved:

- 87.3% accuracy in identifying gender-biased content
- 82.6% precision and 79.4% recall for detecting subtle gender stereotypes
- 91.2% accuracy for flagging explicit discriminatory language

Figure 8 shows the results of the text classification evaluation. These results demonstrate the potential of automatic text classification for enhancing gender equality in educational materials.

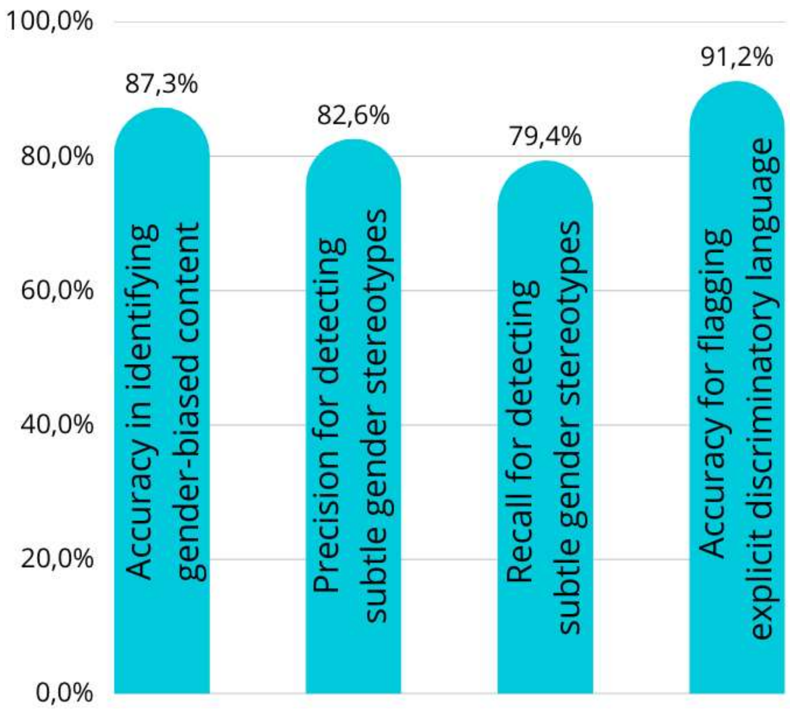

Figure 8: Text classification evaluation.

Integration Capabilities Testing

We tested the system's integration capabilities by connecting it with three external systems:

- A metaverse learning environment
- A third-party assessment tool
- A video conferencing platform

Integration success was measured by data consistency, user experience continuity, and technical stability. Our document-oriented architecture achieved seamless integration with all three systems, maintaining 99.2% data consistency and requiring minimal custom code development compared to traditional approaches (see Fig. 9).

Figure 9: Integration capabilities testing.

7. Conclusion

This paper presents a novel approach to e-learning through a document-oriented Learning Management System and provides answers to our initial research questions. Regarding the first question on enhancing LMS flexibility, our experimental results demonstrate that a document-oriented architecture reduces adaptation time by 68% compared to traditional systems, enabling rapid response to changing educational needs in crisis situations. For the second question on text classification benefits, we found that automatic analysis of educational materials achieves 87.3% accuracy in identifying gender-biased content, providing a powerful tool for promoting gender equality in education. Addressing the third question on integration techniques, our system demonstrated seamless connectivity with diverse learning modalities, maintaining 99.2% data consistency across integrated platforms. These findings confirm that our three-pillar approach – flexible architecture, text classification capabilities, and advanced integration – forms a comprehensive solution that significantly enhances the effectiveness of distance learning in challenging circumstances.

A review of related works underscores the transformative potential of advanced LMS systems while identifying prevalent gaps in integration and functionality. Our proposed LMS framework addresses these deficiencies by offering a structured yet adaptable platform that supports many learning activities and assessment types.

The detailed ontology of system components emphasizes the need for a coherent structure in LMS development, aligning educational materials and assessments with specific learning objectives to facilitate a logical progression through learning and course units, thus enhancing both teaching and learning processes.

The system architecture is crafted to seamlessly integrate diverse e-learning components, ensuring comprehensive support and positioning the framework as a case study for future development. Integrating automatic text classification methods into an LMS offers excellent potential to improve the efficiency and personalization of learning. The proposed approaches can be used to automate the assessment of student work, personalize learning, and integrate with external resources. Further research will focus on refining individual components, such as short answer assessments, learning path recommendations, and course ranking, to improve the learning effectiveness and user experience.

Acknowledgements

The EU NextGenerationEU partially funds the research study depicted in this paper through the Recovery and Resilience Plan for Slovakia under project No. 09I03-03-V01-00078.

Declaration on Generative AI

During the preparation of this work, the author(s) used Grammarly in order to: Grammar and spelling check. Further, the author(s) used DeepL: Text translation. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] H. Ulferts, I. Hu, A. Echazarra, Learning During – and From – Disruption, volume 2 of PISA 2022 Results, 2023. doi:10.1787/a97db61c-en.

[2] Kilag, O. K., Obaner, E., Vidal, E., Castañares, J., Dumdum, J. N., & Hermosa, T. J.: Optimizing Education: Building Blended Learning Curricula with LMS. Excellencia: International Multi-Disciplinary Journal of Education (2994-9521) 1.4 (2023) 238–250.

[3] Ghouali, K., Benmoussat, S., Ruiz-Cecilia, R., E-assessment on the spotlight: present and future prospects. ReiDoCrea: Revista electrónica de investigación Docencia Creativa 9 (2020) 52–62. doi: 10.30827/Digibug.59151.
[4] Liu, C., Zhang, H., Zhang, J., Zhang, Z., Yuan, P., Design of a Learning Path Recommendation System Based on a Knowledge Graph. International Journal of Information and Communication Technology Education 19 (2023) 1–18. doi:10.4018/IJICTE.319962.
[5] O. Viberg, M. Khalil, M. Baars, Self-Regulated Learning and Learning Analytics in Online Learning Environments: A Review of Empirical Research, in Proceedings of the 10th International Learning Analytics and Knowledge, Frankfurt, 2020, pp. 524–533. doi:10.1145/3375462.3375483.
[6] S. Alshammari, M. B. Ali, M. S. Rosli, LMS, CMS and LCMS: The confusion among them. Science International 30 (2018) 455–459.
[7] H. Simanullang, J. Rajagukguk, Learning Management System (LMS) Based On Moodle To Improve Students Learning Activity, Journal of Physics: Conference Series 1462 (2020) 012067. doi:10.1088/1742-6596/1462/1/012067.
[8] S. Ninoriya, P. Chawan, B. Meshram, CMS, LMS and LCMS for eLearning. International Journal of Computer 8 (2011) 644–647.
[9] S. Yassine, S. Kadry, M. Sicilia, Learning Analytics and Learning Objects Repositories: Overview and Future Directions, in: Learning, Design, and Technology, Springer International Publishing Switzerland, 2017, pp. 1–30. doi:10.1007/978-3-319-17727-4_13-2.
[10] N. Kasim, F. Khalid, Choosing the Right Learning Management System (LMS) for the Higher Education Institution Context: A Systematic Review, International Journal of Emerging Technologies in Learning (iJET) 11 (2016) 55. doi:10.3991/ijet.v11i06.5644.
[11] D. Malikin, I. Kyrychenko, Research of methods for practical educational tasks generation based on various difficulty levels, in: Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2022), volume 1, Poland, 2022, pp. 1030–1042. URL: https://ceur-ws.org/Vol-3171/.
[12] N.A. Dahan, M. Al-Razgan, A. Al-Laith, M. Alsoufi, M. Alasaly, T. Alfaqih, Metaverse Framework: A Case Study on E-Learning Environment (ELEM), Electronics, 11, (2022), 1616. doi:10.3390/electronics11101616.
[13] N. Sharonova, I. Kyrychenko, G. Tereshchenko, Application of Big Data Methods in E-Learning Systems, in: Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), volume 1, Kharkiv, 2021, pp. 1302–1311. URL: https://ceur-ws.org/Vol-2870/.
[14] R. Smelyakov, A. Chupryna, D. Darahan, S. Midina, Effectiveness of modern text recognition solutions and tools for common data sources, in: Proceedings of the 5th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2021), volume 1, Kharkiv, 2021, pp. 154–165. URL: https://ceur-ws.org/Vol-2870/.
[15] O. Byzkrovnyi, K. Smelyakov, A. Chupryna, L. Savulionienė, P. Sakalys, Comparison of potential road accident detection algorithms for modern machine vision system, Environment. Technologies. Resources. Proceedings of the International Scientific and Practical Conference, 3 (2023) 50–55. doi:10.17770/etr2023vol3.7299.
[16] N. Khairova, Y. Kupriianov, A. Vorzhevitina, O. Shanidze, Models for effective categorization and classification of texts into specific thematic groups (using gender and criminal themes as examples, in: Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Systems (COLINS-2024), volume 4, Lviv, 2024, pp. 37 – 49. URL: https://ceur-ws.org/Vol-3722/.
[17] Y. Kalagin, O. D. Shanidze, Typolohiia ukrainskykh sotsialnykh praktyk immihrantiv 2022 roku (hendernyi aspekt), in: Proceedings of the 2th Mizhnarodna naukovo-praktychna konferentsiia future of science: innovations and perspectives, Stokholm, 2023, pp. 21-27.