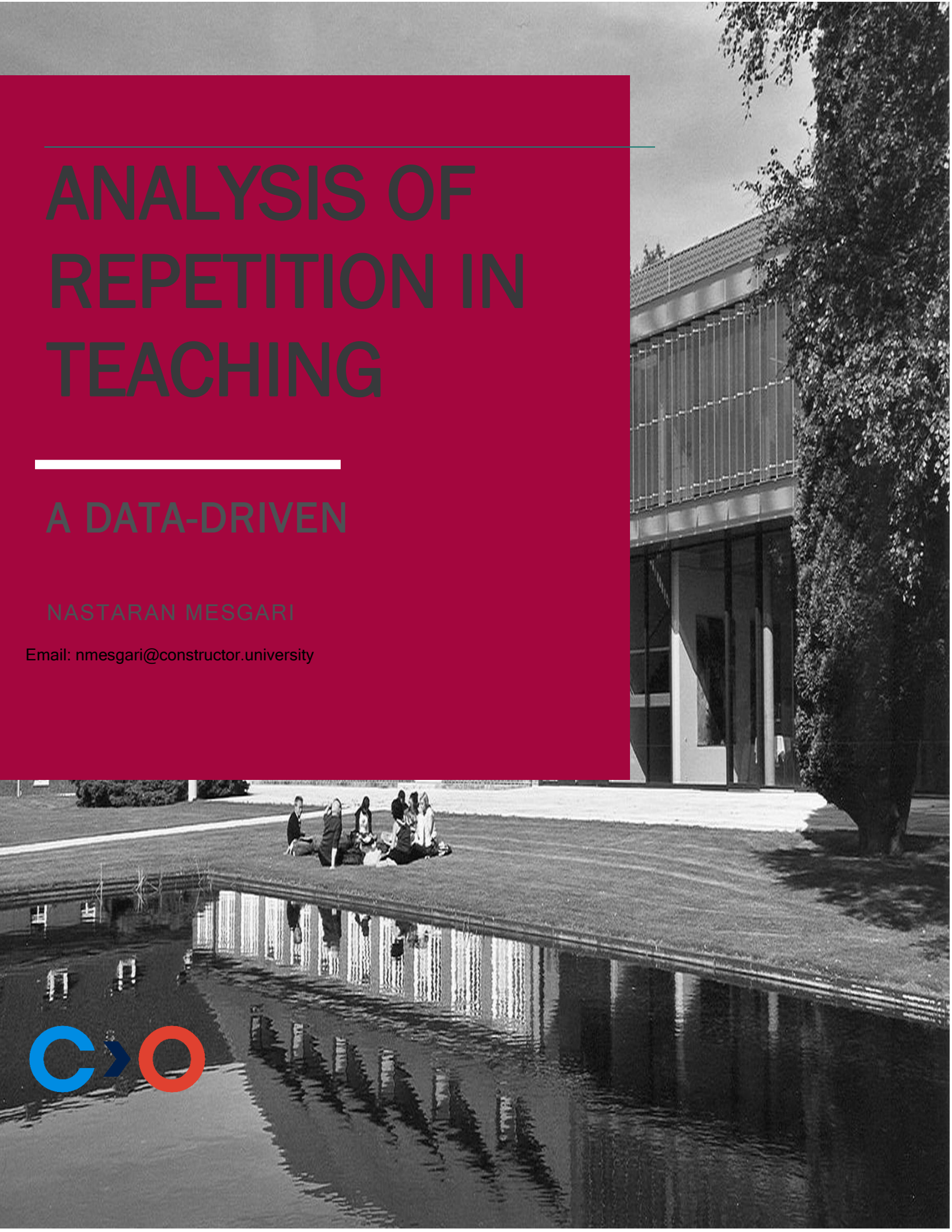
ANALYSIS OF REPETITION IN TEACHING

A DATA-DRIVEN

NASTARAN MESGARI

Email: nmesgari@constructor.university



Introduction and Project Goal

This project aims to analyze content similarities across various courses, with a particular focus on the topics of "Artificial Intelligence" and "Data Visualization." Given the broad scope of these topics and their increasing significance in both academic and industrial contexts, optimizing educational content and preventing redundant material can significantly improve the quality of teaching and students' learning experience. Therefore, this project is designed to identify potential overlaps and provide a framework for more precise and coordinated course structuring in multiple stages.

This project is significant in several ways:

- Optimization of Educational Content: By identifying unnecessary overlaps and repetitions, courses can be structured with greater precision, making the educational content more efficient.

- Increased Teaching Efficiency: It enables greater collaboration among instructors, allowing them to present content in a more coordinated and complementary manner.

- Improved Student Learning Experience: By reducing content repetition, students will have more opportunities to learn new and practical concepts.

- Optimized Educational Resources: Identifying overlaps will lead to better allocation of time and resources.

- Curriculum Development: The findings of this analysis can help in more precise course design and updates.

Proposed Process and Project Implementation Stages

1. Creating a Course List for Analysis

The first step is to compile a comprehensive list of related courses, including:

- Mandatory Core Courses: Courses that all students are required to take, covering foundational topics in Artificial Intelligence and Data Visualization.

- Elective Courses Related to AI and Data Visualization: Courses such as Machine Learning, Natural Language Processing, and Visual Data Analysis.

- Related Courses from Other Disciplines: Such as statistical and mathematical topics that complement the project's themes.

2. Assessing Access to Course Content

In this stage, we determine which courses are accessible:

- Using Completed Courses: Collect and organize files and notes from the courses taken.

- Requesting Official Access: Request formal access to other courses from instructors and relevant authorities.

- Open Educational Resources: Searching for freely available online resources related to the subjects.

- Collaboration with Other Students: Coordinating with fellow students to access different materials.

3. Collecting Accessible Files and Developing the Initial Prototype

In this phase, the collected files will be categorized and processed to develop an initial prototype:

- Categorizing Files: Organizing and prioritizing resources based on course and content type.

- Data Preprocessing: Converting files into text format and removing irrelevant data.

- Developing Initial Algorithm: Designing and implementing an algorithm to identify content similarities and compare materials.

- Creating a User Interface: Designing a basic interface to display results and facilitate easy access to the data.

4. Developing the Prototype and Expanding Access to Data

In this step, the initial prototype will be refined, and efforts to expand access to more resources will continue:

- Optimizing the Algorithm: Enhancing the algorithm's performance and accuracy based on initial feedback.

- Adding New Features: Incorporating functionalities like comparing multiple courses simultaneously and generating more detailed reports.

- Continuing Discussions with the University: To complete access to additional data.

- Collecting Feedback: Gathering input from instructors and students to refine the prototype and algorithms.

Future Stages and Suggested Options

1. Expanding the Prototype with Available Data

Using the available resources, the prototype will be developed as follows:

- Enhancing NLP Algorithms: Using more advanced techniques like Word Embeddings and Topic Modeling for increased accuracy.

- Adding Image Analysis: Comparing the visual similarities of slides and charts.

- Developing an Interactive Dashboard: For a better, user-friendly display of results.

2. Integrating the Prototype into the LMS System

This option aims to integrate the prototype with a Learning Management System (LMS), which includes:

- Designing an API: To connect the prototype with the LMS.

- Creating Automatic Alerts: To notify instructors about content similarities.

- Developing Advanced Reporting: To track changes and similarities over time.

Techniques Used for Identifying Similarities

1. Text Preprocessing

- Stop Word Removal: Removing words that don't carry significant meaning.
- Stemming: Reducing words to their root forms.
- Text Normalization: Converting characters to lowercase and standardizing structure.

2. NLP Algorithms and Similarity Analysis

- TF-IDF: To determine the relative weight of words.
- Word Embeddings: To examine semantic similarity.
- LDA and Topic Modeling: To identify key topics in the content.

3. Similarity Detection Models

- Cosine Similarity and Jaccard Index: To compare similarity between texts.
- Embedding-based Similarity: Using pre-trained models like BERT to extract meaning and context.

Conclusion and Future Steps

This project is designed to identify and reduce content repetition across courses, focusing on improving the learning experience. As the project progresses, further reviews will be conducted to develop, optimize, and implement solutions to create a more effective educational environment.