Software Requirements Specification

For

Search Engine Crawler and Indexer of Mathematical Formulas

Prepared by

Name	SAP ID	Specialization
Bhavya Agrawal	500108017	AIML Hons.
Siddharth Joshi	500107461	AIML Hons.
Sumit Verma	500104649	AIML Hons.

Department of Informatics
School Of Computer Science
UNIVERSITY OF PETROLEUM & ENERGY STUDIES,
DEHRADUN- 248007. Uttarakhand

Mentor Name: Dr. Pankaj Kundan Dadure
Mentor Signature

Table of Contents

Topic		Page No
Table of Content		2
1	Introduction	3
	1.1 Purpose of the Project	3
	1.2 Target Beneficiary	3
	1.3 Project Scope	4
2	Project Description	5
	2.1 Reference Algorithm	5
	2.2 Data/ Data structure	5
	2.3 SWOT Analysis	5
	2.4 Project Features	6
	2.5 Design diagrams	6
	2.6 Assumption and Dependencies	7
3	System Requirements	8
	3.1 User Interface	8
	3.2 Software Interface	8
4	Non-functional Requirements	9
	4.1 Performance requirements	9
	4.2 Security requirements	9
	4.3 Software Quality Attributes	10
5	Other Requirements	11
6	References	11
Appendix A: Glossary		12
Appendix B: Analysis Model		13
Appendix C: Issues List		15

1 INTRODUCTION

1.1 Purpose of the Project

Mathematical formulas are crucial for conveying complex concepts across disciplines. However, discovering specific formulas in digital documents is challenging, particularly with Presentation MathML, which represents the visual structure of formulas. Current search engines and indexing tools often fail to process MathML effectively.

1.2 Target Beneficiary

The target beneficiaries of this project include:

1. **Researchers and Academics**:
    - **Use Case**: Researchers in fields like mathematics, physics, engineering, computer science, and other technical domains frequently need to reference complex mathematical formulas from large bodies of literature. This tool will allow them to quickly locate relevant formulas across vast numbers of articles, improving their research efficiency.
    - **Benefit**: Instead of manually searching through lengthy documents, they can access formulas directly through the indexed and ranked system, saving time and effort.
2. **Educators and Students**:
    - **Use Case**: Professors, teachers, and students often require access to specific mathematical formulas to support teaching, learning, or solving problems.
    - **Benefit**: This system will enable them to quickly find formulas in Presentation MathML format, which is easier to read, understand, and utilize for educational purposes. It enhances teaching materials and provides reliable references for academic studies.
3. **Data Scientists and Engineers**:
    - **Use Case**: Professionals working with data, algorithms, machine learning, or computational modeling often need access to precise mathematical formulations that underpin theoretical models.
    - **Benefit**: This tool will assist them by indexing formulas relevant to their domain, offering easy access to accurate mathematical data needed for building and validating models.
4. **Publishers and Librarians**:
    - **Use Case**: Academic publishers and digital libraries aim to provide readers with easily accessible, well-organized information, including mathematical content.
    - **Benefit**: This project will offer an automated system to sort and organize mathematical formulas in a searchable, indexed manner, improving the accessibility of scholarly articles for readers.
5. **Developers of Mathematical Software**:
    - **Use Case**: Developers creating mathematical, scientific, or educational software often need access to various formulas and algorithms.

- **Benefit**: They can incorporate these indexed and ranked formulas directly into their applications, enhancing the functionality and accuracy of software products.

Overall, the project will benefit any individual or organization that regularly deals with large collections of scientific literature containing mathematical formulas, improving both access to and utilization of complex mathematical information.

1.3 Project Scope

In this project, we aim to crawl through many articles which have in total lakhs of mathematical formulas to search from. The crawler crawls through all these formulas and shows them in the Presentation MathML view. It will create an index list of the most suitable formulas and sort them based on their relevance.

2 PROJECT DESCRIPTION

2.1 Reference Algorithm

The reference algorithm for this project utilizes a rule-based approach for retrieving and indexing mathematical information from scientific documents. The system follows predefined rules to extract mathematical formulas, convert them into binary vectors using Presentation MathML, and capture the surrounding context for accurate interpretation. It then applies a rule-based relevance measurement technique that ranks documents based on the structural similarity of the formulas and their context, improving search accuracy. The system efficiently indexes the formulas and their contexts, enabling fast and precise retrieval without relying on formula generalization, focusing instead on exact or closely matching formulas. This approach enhances the overall performance of Mathematical Information Retrieval (MIR) systems by ensuring consistency, customization, and efficiency.

2.2 Characteristics of Data

The data used is collected from Wikipedia. It is NTCIR data collected from many Wikipedia articles. It contains many Mathematical articles as well as text articles. The math articles are html webpages. The dataset includes digital documents with mathematical content in Presentation MathML. Input formats are diverse, including HTML web pages, scientific papers, mathematical articles and technical documents embedding MathML-encoded expressions.

2.3 SWOT Analysis

Strengths

- Specialized Focus: Tailored for Presentation MathML, enhancing search accuracy.
- Java Advantages: Cross-platform compatibility and robust libraries.
- Advanced Search: Features like structural similarity and sub-formula matching.

Weaknesses

- Encoding Complexity: Challenges in processing various mathematical encodings.
- Normalization Issues: Difficulty in standardizing formulas without losing meaning.
- Scalability: Potential performance issues with large data volumes.

Opportunities

- Growing Demand: Increasing need for specialized mathematical search engines.
- Academic Adoption: Potential use by researchers and institutions.
- Integration: Opportunity to connect with academic databases and platforms.

Threats

- Competition: Risk of superior competing tools.
- Performance Bottlenecks: Scalability issues could affect adoption.
- Ambiguity Challenges: Risks in handling contextual and semantic similarity.

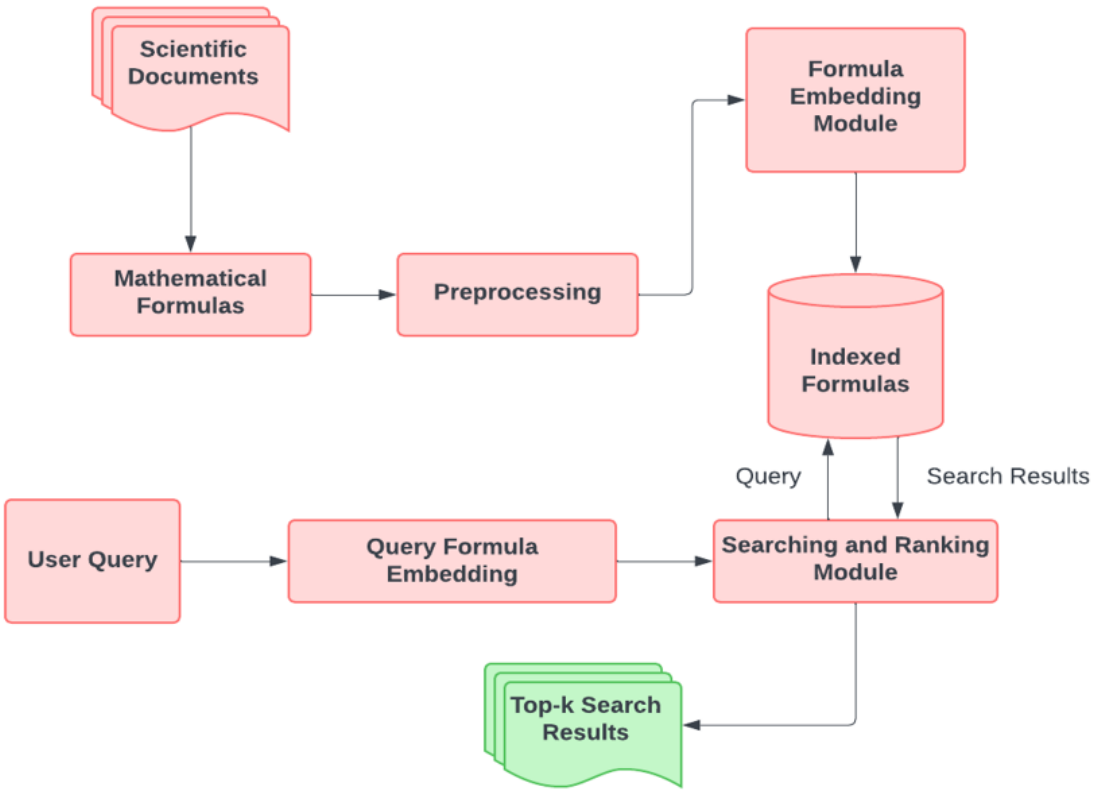
2.4 Project features

This project focuses on developing an advanced web crawler tailored to process large-scale collections of articles containing mathematical formulas. Key features of the project include:

1. **Crawling and Extraction**: The crawler will systematically navigate through articles and extract all mathematical formulas embedded within them.
2. **Presentation MathML Display**: All extracted formulas will be converted and displayed using the Presentation MathML format, ensuring a uniform, machine-readable mathematical representation.
3. **Relevance-based Indexing**: The crawler will analyze and create an index of the most suitable formulas based on predefined relevance criteria, aiding in the identification of the most significant or frequently occurring mathematical expressions.
4. **Sorting and Ranking**: Formulas will be sorted and ranked by their relevance, allowing users to easily locate the most pertinent mathematical information from a vast collection of data.
5. **Efficiency and Scalability**: The system is designed to handle large datasets, potentially processing thousands or even lakhs (hundreds of thousands) of mathematical formulas efficiently.

These features aim to streamline the retrieval of important mathematical information, making it easier for researchers and professionals to access and work with complex mathematical data across a wide array of articles.

2.5 Design Diagrams



2.6 Assumptions and Dependencies

The project described in the text has several assumptions and dependencies that affect its design, implementation, and performance.

Assumptions:

1. **Availability of Mathematical Information in Presentation MathML Format**: It is assumed that the documents to be processed contain formulas that can be extracted in Presentation MathML format, which is crucial for consistency in displaying and interpreting mathematical content.
2. **Binary Representation of Formulas**: The proposed method assumes that formulas can be effectively represented as binary vectors (where '1' indicates the presence of an entity and '0' its absence). This assumes that formula structure and components can be sufficiently captured by this binary model.
3. **Combined Relevance of Formula Embedding and Generalization**: It is assumed that a combination of formula embedding and generalization modules yields better retrieval results compared to using them independently. The relevance measurement technique is based on this assumption.
4. **Test Data Generalizability**: The approach has been tested on MathTagArticles of Wikipedia (NTCIR-12 dataset). It is assumed that the performance metrics obtained on this dataset are representative and that the method will generalize well to other collections of scientific documents.

Dependencies:

1. **Document Preprocessing Module**: The success of the approach is dependent on the document preprocessor module, which is responsible for extracting the formulas in Presentation MathML format along with their context. This preprocessing step is critical to ensure that the formulas are correctly identified and extracted.
2. **Formula Embedding**: The project relies heavily on the formula embedding. This module creates binary vectors for each formula, which are used for indexing and retrieval. The accuracy of this module directly impacts the relevance and quality of the search results.
3. **Innovative Relevance Measurement**: The method uses a novel relevance measurement technique, which ranks documents based on both formula embedding and generalization. The system depends on this technique to provide improved search accuracy. The relevance model must be robust enough to handle various types of mathematical data.
4. **Indexing Mechanism**: The approach depends on the efficiency and accuracy of the indexing mechanism to store the binary vectors of formulas. The indexer plays a key role in enabling fast and relevant search queries.
5. **Dataset and Benchmarking**: The results and evaluation are dependent on the NTCIR-12 MathTagArticles dataset. This is used as the benchmark for testing the system, meaning performance improvements are contingent on the quality and structure of this dataset.

3 SYSTEM REQUIREMENTS

Hardware Requirements:

- **Processor:** Multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5) for handling concurrent tasks.
- **RAM:** Minimum 8 GB (16 GB recommended) for processing large files.
- **Storage:** At least 500 GB (SSD recommended) for storing documents and indexed data.
- **Network:** Internet connection for fetching documents if needed.

Software Requirements:

- **Operating System:** Compatible with Windows, macOS (10.13 or higher), or Linux.
- **Java Development Kit (JDK):** Version 8 or higher.
- **Java IDE:** IntelliJ IDEA, Eclipse, or Visual Studio Code for development.
- **Libraries:** Built-in Java libraries (java.io, java.util.regex).
- **File Handling Tools:** Tools or utilities for handling file operations, such as Unix command line utilities (grep, sed, awk), can be useful for pre-processing or batch handling large sets of documents.

4 NON-FUNCTIONAL REQUIREMENTS

4.1 Performance Requirements

The performance requirements for the project, which focuses on a rule-based approach to Mathematical Information Retrieval (MIR) systems, can be categorized into several key areas:

1. **Accuracy:** The system should achieve high accuracy in retrieving relevant mathematical formulas based on user queries. This includes correctly identifying and indexing formulas and their contexts to ensure that the most pertinent results are ranked highest.
2. **Precision and Recall:** The system should return a high proportion of relevant results among all retrieved documents. The relevance measurement technique must minimize false positives. The system should be capable of retrieving a significant percentage of all relevant documents available in the dataset, ensuring that users do not miss important mathematical information.
3. **Processing Speed:** The formula extraction, indexing, and retrieval processes should be efficient, enabling the system to handle large datasets (potentially containing lakhs of formulas) within a reasonable time frame. Users should receive results quickly, ideally within seconds of submitting a query.
4. **Scalability:** The system should be able to scale effectively as the volume of documents and mathematical formulas increases. It should maintain performance levels without degradation in speed or accuracy when processing larger datasets.
5. **Robustness:** The system should handle variations in document formats and structures effectively. It should be resilient to common issues such as noise in the data, formatting inconsistencies, or incomplete formulas, ensuring reliable performance across diverse sources.
6. **Contextual Relevance:** The performance should include the ability to accurately interpret and leverage the context surrounding formulas. This means that the system should effectively utilize the contextual information to improve retrieval accuracy and relevance.
7. **User Experience:** The system should provide a user-friendly interface that allows for easy input of queries and quick navigation through search results. It should also support various query types, enabling users to search for specific formulas or broader mathematical concepts.
8. **Maintenance and Update Frequency:** The system should allow for regular updates to the underlying dataset and indexing rules without significant downtime. This ensures that users have access to the most current mathematical information.

4.2 Security Requirements

Here are the security requirements for the project focused on Mathematical Information Retrieval (MIR) systems:

1. **Data Protection:** Ensure that all sensitive data, including formulas and user queries, are encrypted both in transit and at rest to prevent unauthorized access and data

breaches. Implement strict access control measures to ensure that only authorized users can access or modify the system and its data.

2. **User Authentication:** Utilize secure authentication methods (e.g., multi-factor authentication) to verify the identity of users accessing the system, protecting against unauthorized access.
3. **Data Integrity:** Implement checks to ensure that the data extracted, stored, and processed remains accurate and unaltered. This can include checksums or hashing techniques to verify data integrity.

4.3 Software Quality Attributes

Here are the **software quality attributes** for the Mathematical Information Retrieval (MIR) system:

1. **Performance:** The system should provide quick responses to user queries, ideally within seconds, even when processing large datasets. The ability to handle multiple simultaneous queries efficiently without degrading performance.
2. **Usability:** The system should have an intuitive and user-friendly interface that allows users to easily input queries and navigate search results. Ensure that the system is accessible to users with disabilities, following relevant guidelines and standards.
3. **Reliability:** The system should be resilient to failures, maintaining functionality and providing accurate results even in the event of errors or system interruptions. Ensure high availability of the system, minimizing downtime for users.
4. **Maintainability:** The system should be designed in a modular way, allowing for easier updates and maintenance of individual components without affecting the entire system. Maintain high code quality standards, including documentation and adherence to coding conventions, to facilitate future modifications.
5. **Scalability:** The system should be capable of scaling horizontally (adding more servers) or vertically (upgrading existing hardware) to accommodate increased workloads as the dataset grows.
6. **Extensibility:** The architecture should allow for easy addition of new features or functionalities without significant rework of existing components.

5 OTHER REQUIREMENTS

Training Requirements:

- **User Training**: Training sessions should be conducted for users to familiarize them with the system's functionalities and interface.
- **Technical Training**: Developers and system administrators should receive training on system maintenance, updates, and troubleshooting.

Support Requirements:

- **Technical Support**: A support mechanism should be in place to assist users with issues and queries related to the system.
- **Maintenance Schedule**: Regular maintenance and updates must be planned to ensure system reliability and performance.

6 REFERENCES

[1] Gao, L., Jiang, Z., Yin, Y., Yuan, K., Yan, Z., & Tang, Z. (2017). Preliminary Exploration of Formula Embedding for Mathematical Information Retrieval: Can mathematical formulae be embedded like a natural language? (arXiv:1707.05154). arXiv. http://arxiv.org/abs/1707.05154

[2] Pathak, A., Pakray, P., & Das, R. (2019). LSTM Neural Network Based Math Information Retrieval. 2019 Second International Conference on Advanced Computational and Communication Paradigms (ICACCP), 1–6. https://doi.org/10.1109/ICACCP.2019.8882887

[3] Pathak, A., Pakray, P., Sarkar, S., Das, D., & Gelbukh, A. (2017). MathIRs: Retrieval System for Scientific Documents. Computación y Sistemas, 21(2). https://doi.org/10.13053/cys-21-2-2743

[4] Polewczak, J. (n.d.). LATEX, MATHML, AND TEX4HT: TOOLS FOR CREATING ACCESSIBLE DOCUMENTS. http://www.csun.edu/~hcmth008/mathml/acc_tutorial.pdf

[5] Sojka, P., Růžička, M., & Novotný, V. (2018). MIaS: Math-Aware Retrieval in Digital Mathematical Libraries. Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 1923–1926. https://doi.org/10.1145/3269206.3269233

Appendix A: Glossary

- **Algorithm**: A set of rules or steps used to solve a specific problem or perform a computation, such as retrieving mathematical information from documents.
- **Binary Vector**: A representation of data using a vector composed of binary values (0s and 1s), where each position indicates the presence (1) or absence (0) of a specific entity.
- **Contextual Analysis**: The process of evaluating the surrounding text or information related to a mathematical formula to enhance understanding and relevance during retrieval.
- **Data Protection Regulations**: Legal frameworks, such as GDPR or HIPAA, that govern the handling and protection of personal and sensitive information.
- **Document Preprocessor**: A module that prepares documents for analysis by extracting mathematical formulas and converting them into a specific format (e.g., Presentation MathML).
- **Formula Extraction**: The process of identifying and retrieving mathematical formulas from scientific documents.
- **Formula Embedding**: The technique of transforming mathematical formulas into a machine-readable format, often represented as binary vectors.
- **Mathematical Information Retrieval (MIR)**: A field focused on the retrieval of mathematical content and information from documents, including formulas, equations, and related context.
- **Presentation MathML**: A markup language used for displaying mathematical notations and formulas in a structured format that can be rendered by web browsers.
- **Precision**: A measure of the accuracy of retrieved results, calculated as the ratio of relevant results to the total number of retrieved results.
- **Recall**: A measure of the ability of a system to retrieve all relevant results from a dataset, calculated as the ratio of relevant results to the total number of relevant results available.
- **Relevance Ranking**: The process of organizing retrieved documents based on their relevance to a user's query, often determined by contextual and content analysis.
- **Rule-Based Approach**: A methodology that employs predefined rules to guide the extraction, analysis, and retrieval of information.
- **Scalability**: The capability of a system to handle increased workloads, such as larger datasets or more user queries, without a decrease in performance.
- **Security Measures**: Protocols and techniques employed to protect data and systems from unauthorized access, breaches, and other threats.
- **Usability**: The ease with which users can interact with a system, encompassing user interface design and overall user experience.
- **User Interface (UI)**: The means by which a user interacts with a computer system, including visual elements such as buttons, menus, and input fields.
- **Vulnerability Management**: The process of identifying, assessing, and addressing security weaknesses in a system to prevent unauthorized access and data breaches.

Appendix B: Analysis Model

1. Overview:

The MIR system is designed to efficiently extract mathematical formulas from scientific documents, analyze their context, and retrieve relevant information based on user queries. The system employs a rule-based approach for extraction, embedding, and ranking.

2. Components:

A. Input Module:

- **Document Source**:
  - Input from various scientific documents (e.g., PDFs, articles, textbooks).
- **User Queries**:
  - Input queries from users searching for specific mathematical formulas or concepts.

B. Preprocessing Module:

- **Document Preprocessor**:
  - Extracts text and mathematical content from documents.
  - Converts formulas into **Presentation MathML** format.

C. Formula Extraction Module:

- **Rule-Based Extractor**:
  - Identifies mathematical formulas based on predefined rules.
  - Extracts contextual information related to each formula.

D. Embedding Module:

- **Formula Embedding**:
  - Converts extracted formulas into binary vectors representing the presence or absence of specific entities.

E. Indexing Module:

- **Indexer**:
  - Creates an index of extracted formulas and their contextual information.
  - Stores binary vectors for efficient retrieval.

F. Relevance Measurement Module:

- **Ranking Algorithm**:
  - Implements a rule-based approach to rank documents based on the relevance of formulas and their contexts.
  - Combines scores from both formula embedding and contextual analysis.

G. Retrieval Module:

- **Query Processor**:
  - Receives user queries and retrieves relevant documents from the index.
  - Applies ranking to present results in order of relevance.

H. Output Module:

- **Results Display**:
  - Presents the retrieved documents and formulas in a user-friendly format, including a view in **Presentation MathML**.

3. Data Flow:

- **Input**:
  - User queries and documents are input into the system.
- **Processing**:
  - Documents are preprocessed, and mathematical formulas are extracted and embedded into binary vectors.
- **Indexing**:
  - The extracted data is indexed for quick access.
- **Retrieval**:
  - The system processes user queries, retrieves relevant results, and ranks them based on contextual relevance.
- **Output**:
  - Display of results to the user in an understandable format.

4. Interaction Diagrams:

- **Use Case Diagram**:
  - Illustrates user interactions with the system (e.g., submitting queries, viewing results).
- **Sequence Diagram**:
  - Represents the flow of operations from document input through to output display.

5. Performance Metrics:

- **Response Time**:
  - Measure the time taken to retrieve results after a query is submitted.
- **Accuracy Metrics**:
  - Precision and recall rates for retrieved documents.
- **Scalability Testing**:
  - Evaluate system performance under increased data loads.

Conclusion:

This analysis model outlines the key components, processes, and interactions within the MIR system. By implementing this model, the project aims to enhance the efficiency and effectiveness of retrieving mathematical information from diverse scientific documents, ultimately improving user experience and accessibility to mathematical knowledge.

Appendix C: Issues List

1. Ambiguity in Requirements:
- Some requirements may be vague or open to interpretation, leading to confusion during development. Clear definitions and examples may be needed to ensure all stakeholders have a common understanding.

2. Evolving User Needs:
- As users interact with the system, their needs may evolve, leading to new requirements that were not initially captured. A flexible approach to requirement gathering and prioritization is necessary to adapt to these changes.

3. Integration Challenges:
- Integrating the MIR system with existing tools and databases may pose challenges. Compatibility issues and varying data formats can complicate the integration process, requiring additional requirements for data transformation and API development.

4. Performance Expectations:
- There may be differing expectations regarding system performance, particularly regarding response times and throughput. Clearly defined performance metrics and targets should be established to align stakeholder expectations.

5. Security and Compliance Requirements:
- Security and compliance requirements, such as data protection regulations, may not be fully understood or documented. Ensuring compliance with regulations like GDPR or HIPAA may introduce additional complexity to system design and implementation.

6. Usability Concerns:
- Users may have diverse backgrounds and levels of expertise, leading to differing usability expectations. Conducting user research and testing is essential to identify usability issues and inform design decisions.

7. Testing and Validation Issues:
- Establishing comprehensive testing strategies to validate functional and non-functional requirements may be challenging. Open issues may include determining the extent of automated vs. manual testing and the criteria for success.

8. Resource Constraints:
- Limited resources, such as time, budget, or personnel, may hinder the ability to fully address all requirements. Prioritization of requirements may be necessary, but this can lead to open issues regarding which features or functionalities to include.

9. Documentation Gaps:
- Incomplete or unclear documentation of requirements may result in misunderstandings among team members and stakeholders. Ongoing documentation and communication efforts are essential to mitigate this risk.

10. Maintenance and Support Requirements:
- There may be uncertainty regarding ongoing maintenance and support needs for the system post-implementation. Defining these requirements early on can help ensure the system remains functional and up-to-date.