INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY BANGALORE

NATURAL LANGUAGE PROCESSING AI-829

MATHIQ

Aditi Singh
MT2023085
aditi.singh@iiitb.ac.in

Parag Dutt Sharma
MT2023095
parag.sharma@iiitb.ac.in

1. Project Description

Our project, MathIQ, aims to develop a system that translates natural language mathematical problems, i.e, word problems into equations for accurate solutions. Using NLP techniques, the model will understand diverse mathematical scenarios including word problems across various domains of mathematics. By converting user-provided mathematical word problems into mathematical expressions, the system ensures a transition from linguistic input to mathematical equations, serving educational, practical, and analytical needs. The project aims to enhance problem-solving capabilities, making complex mathematical tasks easier to understand through natural language processing.

2. Project Importance

Accessibility and Education: MathIQ makes mathematical concepts more accessible to a variety of users from different domains. It can be particularly helpful for students who may struggle with understanding traditional mathematical notation but are able to understand the English language.

Problem Solving Enhancement: MathIQ can enhance problem-solving capabilities by converting the word problems into their corresponding mathematical equations. Users can focus on understanding the problem conceptually, and MathIQ will handle mathematical representation of it.

Cross-Domain Applicability: MathIQ aims to understand diverse mathematical problems for various domains. This versatility allows the system to be helpful for a wide range of problems, making it a valuable tool for both educational and practical purposes.

Time Efficiency: For both students and professionals, converting word problems into mathematical equations manually can be time-consuming. Here comes the work of MathIQ, it provides a time-efficient solution by doing this process for them.

Analytical Assistance: The system's ability to convert word problems into equations can be useful in data analysis and decision-making processes. It allows for quick translation of real-world situations into mathematical equations, providing better decision support.

3. Technologies To Be Used

- **Datasets** from **Kaggle, Google dataset, GitHub**.
- Experiment with various **LLMs (Large Language Models).**
- **TensorFlow** to be used as the **framework** for building and training the machine learning model of MathIQ.
- **NumPy** will be used for **data manipulation** and **numerical calculations** such as storing, organizing, and processing the numerical data
- **Scikit-learn** provides a great library for machine learning algorithms and tools that are compatible with **TensorFlow**.
- **Jupyter Notebook** will be used as the **development environment** for the project, allowing us to write, execute, and document our code.
- **Hugging Face** will be used for deployment.

4. Process Flow

Data Collection and Preprocessing:

- **Data Collection:** Obtaining a dataset of math word problems with corresponding **LaTeX solutions**.
- **Data Filtering and Cleaning:** Cleaning and filtering the dataset like removing errors, inconsistencies, and irrelevant information.
- **Data Augmentation:** Techniques like paraphrasing and synonym substitution might be used to increase the dataset size and improve model versatility.

LLM Exploration:

- Using different LLMs such as **GPT-3, BERT, Bloom, LaMDA, LAMA2** to find the best fit for the domain and task.
- Consider factors like accuracy, response quality, domain-specific knowledge, and computational efficiency.

Model Training and Optimization:

- **Model Architecture Design:** A neural network architecture, possibly based on Transformers, will be designed to handle the natural language processing and symbolic tasks.
- **Model Training:** The model will be trained on the prepared dataset, learning to map text descriptions of math problems to their corresponding LaTeX solutions.
- **Hyperparameter Tuning:** Different model parameters and training configuration settings will be experimented to optimize the accuracy and performance.

Deployment and User Interaction:

- **Model Deployment:** The trained model will be deployed on a platform like Hugging Face, making it accessible through a user interface.
- **User Input:** Users can provide a math word problem as text input into the interface.
- **Natural Language Processing:** The NLP model within the deployed model will then analyze the user input and extracts the relevant mathematical concepts and relationships.
- **Symbolic Representation:** The extracted information will be translated into a LaTeX representation of the problem.
- **Solution Generation:** The model will be able to utilize its trained knowledge to solve the problem represented in LaTeX and generate the corresponding solution.
- **Output Presentation:** The solution will be presented to the user through the interface, possibly in LaTeX format alongside a textual explanation.

Evaluation and Improvement:

- **Model Evaluation:** The model's performance will then be evaluated on a separate test dataset to assess its accuracy and generalization capabilities.
- **Error Analysis:** Common errors and misconceptions identified during evaluation will be analyzed to improve the model's training and performance.
- **Model Refinement:** Based on the evaluation and error analysis, the model architecture, training process, or data pre-processing techniques might be adjusted to improve performance and address identified weaknesses.

5. Milestones To Achieve In Future Mandates

Mandate 2: Improve and consolidate the knowledge base through the implementation of lexical preprocessing techniques. These techniques include stemming, lemmatization, identification of mathematical statements using methods such as CAP, PMI, and N-grams, recognition of problem statement variants, and the utilization of phonetic hashing.

Mandate 3: Utilizing Shallow Parsing and POS Tagging helps establish required grammatical structures in problem statements. The usage of Hidden Markov Models (HMMs) with the Viterbi Heuristic allows for effective modeling of word sequences. This approach helps in understanding the correlation of mathematical terms and managing variations in problem presentation. We will start with using models like BERT, GPT, and LAMA 2.

Mandate 4: Implement Word Sense Disambiguation (WSD) techniques to accurately understand the meanings of ambiguous words in specific mathematical contexts. Integrate Named Entity Recognition (NER) to identify and categorize entities like mathematical terms, symbols, and units in given math problems. Apply Topic Modeling to capture the underlying mathematical terms in user queries. Employ fine-tuning methods and design a user-friendly interface to optimize the performance of MathIQ.

$a^2+b^2=c^2$
$A=\frac{1}{2}AB\sin C=A^2+B^2+C^2$
$27/32=0.8954$
$\frac{\sqrt{3}}{4}=(a^2)$
$\frac{1}{2}+B=0 \Rightarrow B=\frac{1}{2}$
$x^2=10$
%
Σ
π

6. Team Members and Their Role

ADITI SINGH (MT2023085):
Focuses on the technical aspects of model training and optimization.
Handles tasks like:

- Data pre-processing and curation.
- Choosing and configuring the deep learning model architecture.
- Training and hyperparameter tuning of the model.
- Evaluating the model's performance and analyzing errors.

PARAG DUTT SHARMA (MT2023095):
Focuses on the natural language processing and symbolic representation aspects.
Handles tasks like:

- Developing the NLP pipeline for text understanding and problem extraction.
- Implementing the LaTeX encoding and processing functionalities.
- Designing the output presentation format (LaTeX, natural language, or hybrid).
- Creating user interface elements for problem input and solution display.