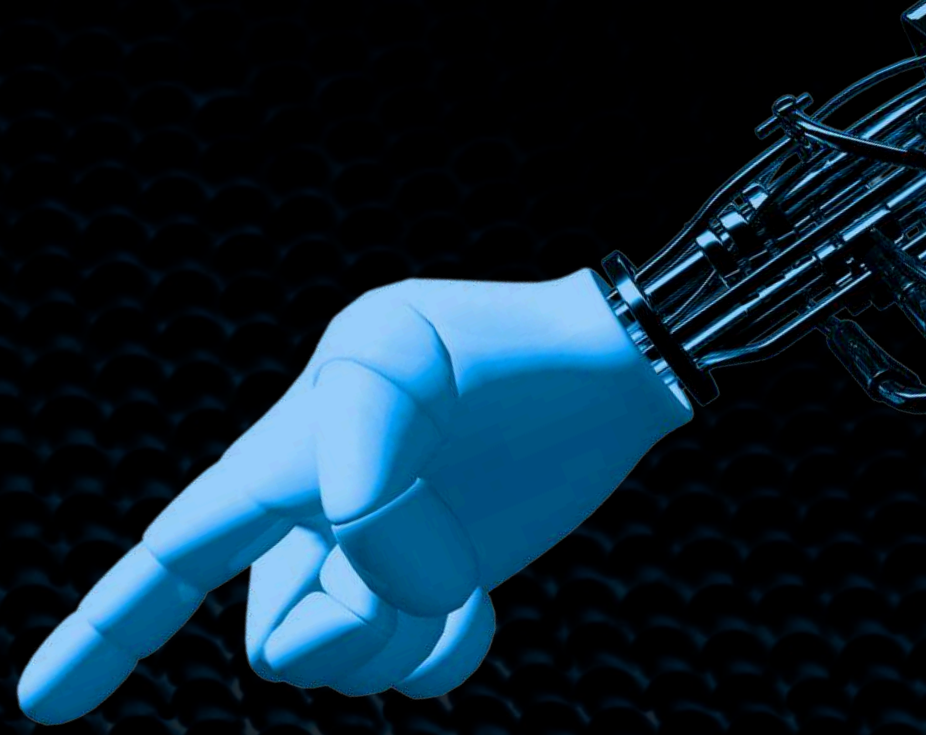
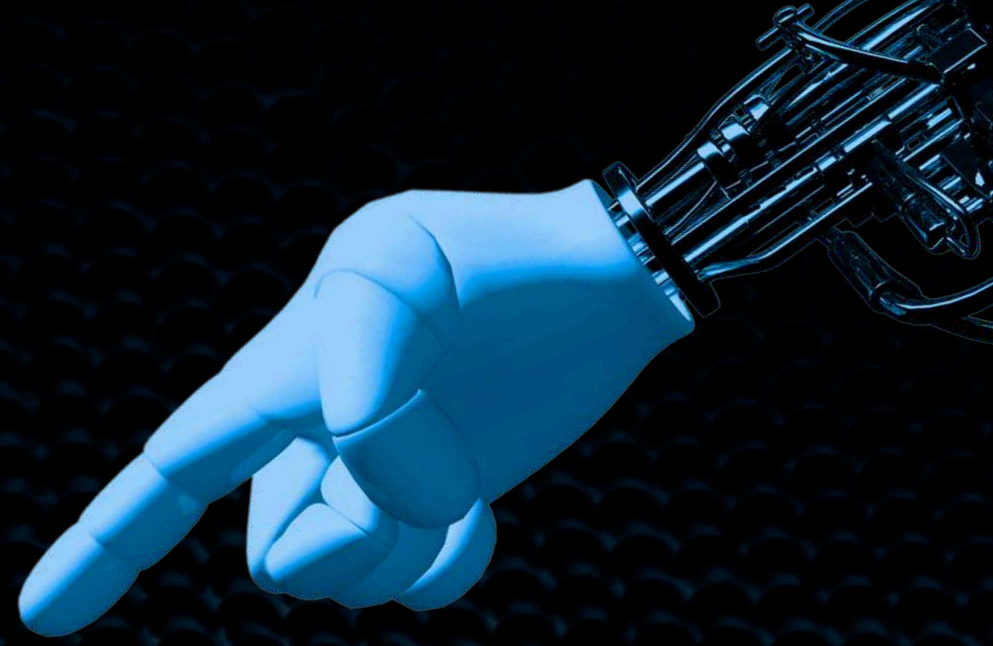
INTELLIGENT SYSTEM FOR DETECTING STUDENT MISCONCEPTIONS IN MATHEMATICS

Steven Navarro Parrales - 20221020048
Juan David Amaya Patino˜ - 20221020057
David Santiago Garcia - 20231020158



Project Overview

The project develops an intelligent system that automatically identifies student misconceptions in mathematics by analyzing their wrong answers in multiple-choice questions. Traditional methods rely on manual teacher interpretation, which is time-consuming and inconsistent.



Project Overview

The principal challenges for our project were:

- Students make systematic errors in mathematical reasoning
- Teachers need automated tools to understand these errors
- Manual misconception labeling is not scalable

Our Solution:

We created a system that uses Natural Language Processing and Large Language Models to automatically detect and classify these misconceptions, helping teachers provide better feedback.

Project Objectives

To design and implement an intelligent system that automatically detects and classifies student misconceptions in mathematics using Large Language Models and NLP techniques.

1

Analyze

Analyze the competition dataset and identify key entities (distractors, misconceptions, constructs)

2

Design

- Design a modular system architecture with semantic filtering and LLM integration

3

Test

- Test system performance across varying data complexities (100, 500, 1000 misconceptions)

4

Integrate

- Integrate Ollama as a local LLM for misconception tagging and validation

5

Generate

- Generate competitive predictions using MAP@25 evaluation metric

6

Submit

1. Submit a working solution to the Kaggle competition

Development Approach

We followed a systems engineering approach combining multiple methodologies

Systems Thinking Integration:

- Modular architecture with interconnected components
- Cybernetic feedback loops for continuous improvement
- Chaos theory principles to handle unpredictable NLP behavior

Development Steps:

1. Competition Analysis - Understanding the problem structure
2. System Design - Creating modular architecture
3. Simulation & Testing - Evaluating across different complexities
4. LLM Integration - Implementing Ollama for validation
5. Submission - Generating final predictions for Kaggle

Proposed Solution Architecture

01 — Trainer Data Module

Processes questions, answers, and misconceptions

02 — Affinity Filter Module

Calculates semantic similarity between distractors and misconceptions

03 — NLP Tagger

Labels distractors with likely misconceptions

04 — LLM Verifier

Uses Ollama to validate and improve predictions

05 — Verification Module

Provides feedback loop for error correction

Technologies and Tools

Programming & Data Processing:

- Python 3.10
- pandas and NumPy
- scikit-learn for machine learning (RandomForest classifier)

NLP & Semantic Analysis:

- SentenceTransformers
- Cosine similarity
- Transformer-based language models

LLM Integration:

- Ollama - Local LLM deployment framework

Data Processing Methodology

Data Preprocessing:

- Text standardization and special character removal
- Tokenization of questions, answers, and misconceptions
- Invalid data filtering and quality control
- Label encoding for multi-class classification

Semantic Similarity Analysis:

- Generated embeddings using SentenceTransformers
- Calculated cosine similarity between distractors and misconceptions
- Applied threshold filtering (0.15) to retain only relevant pairs
- Ranked misconceptions by semantic affinity scores

LLM Integration Strategy

We chose Ollama for local LLM deployment because it provides:

- Complete offline operation capability
- Data privacy and security control
- Cost efficiency compared to cloud APIs
- Consistent, reproducible performance

Integration Points:

1. Post-Filtering Tagging - After semantic filtering, Ollama tags each distractor with relevant misconceptions
2. Validation Loop - The same model verifies tag quality and suggests corrections for low-confidence predictions

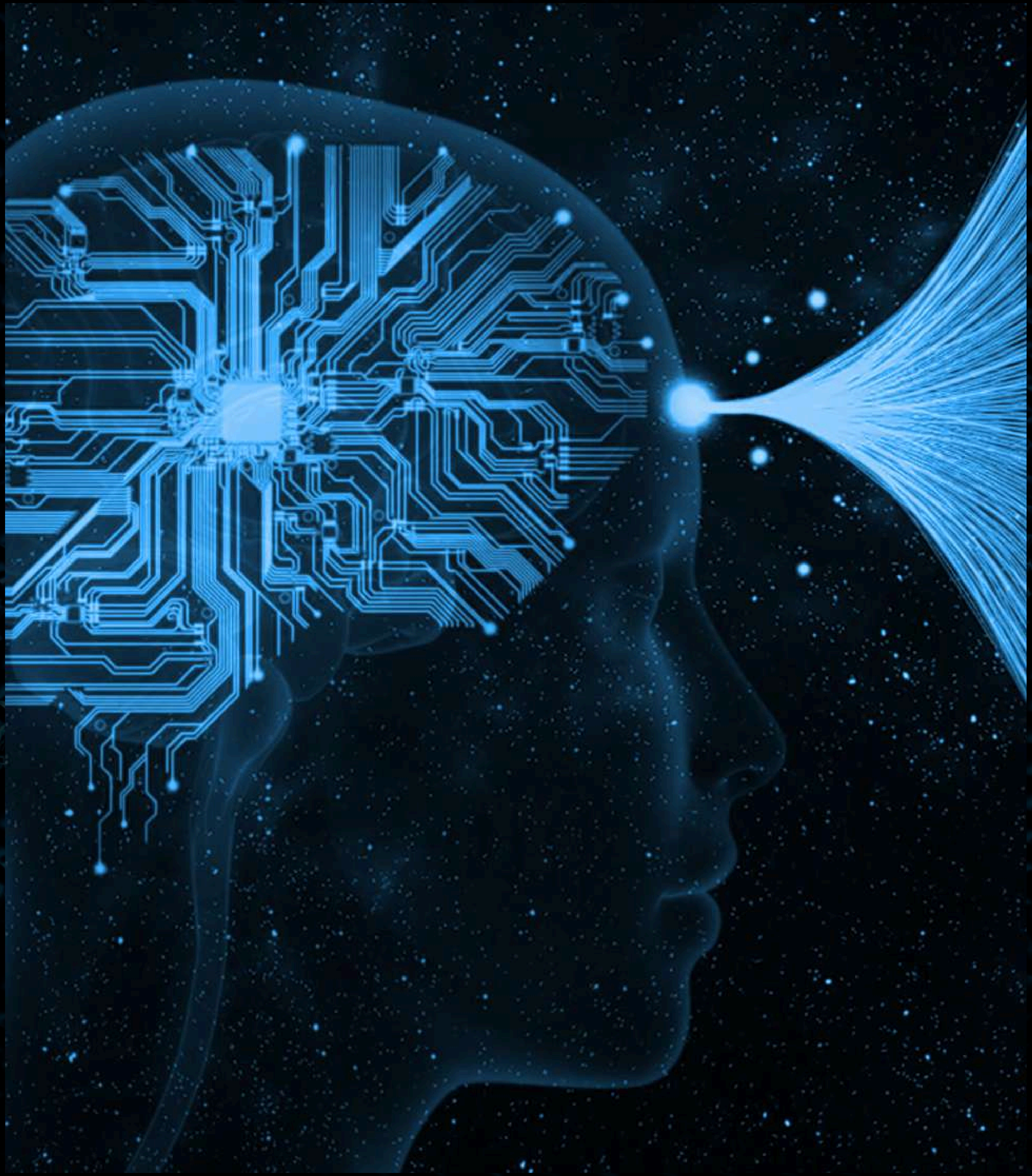
Experimental Design

Dataset Complexity Testing:

- V1: 100 misconceptions (baseline)
- V2: 500 misconceptions (medium complexity)
- V3: 1000 misconceptions (full complexity)

Evaluation Metrics:

- MAP@25 (Mean Average Precision at 25) - Primary competition metric
- Classification accuracy and confusion matrices
- Processing time and system stability
- High-affinity pair detection rates
- LLM rejection rates during validation



Cybernetic Feedback System

Verification and Feedback Loop:
Our system implements cybernetic principles through a sophisticated feedback mechanism:

Validation Process:

1. Initial predictions are assessed for confidence and semantic validity
2. Low-confidence predictions are flagged for reprocessing
3. Ollama provides alternative suggestions or corrections
4. Failed predictions are reinjected into the pipeline

Key Results - Performance Metrics

Primary Achievement:

- MAP@25 Score: 0.811 on the 1000 misconceptions dataset
- Competitive performance meeting Kaggle standards

Classification Accuracy

82% - 100% across different datasets

High-Affinity Pair Rate

73% of filtered distractors

Ollama Rejection Rate

6% during validation

Average Processing Time

1.2 seconds per prompt

System Stability

±0.021 standard deviation across simulations

System Benefits and Impact

Modular Design:

- Independent components support scalability and maintenance
- Easy integration of new LLMs or processing modules
- Parallel processing capabilities for large datasets

Educational Impact:

- Reduces manual effort in misconception identification
- Enables personalized learning at scale
- Provides explainable AI reasoning for teachers
- Supports intelligent tutoring system development

Practical Benefits:

- Offline operation ensures data privacy
- Cost-effective compared to cloud-based solutions
- Fast processing suitable for real-time applications
- Robust performance across varying complexity levels

Limitations and Challenges

Ingeniera Leire Arza | Conferencia de tecnologías en función del mejoramiento humano. Borcelle, 2030.

Domain and Language Constraints:

- System focused on English-language mathematics content
- Limited testing on multilingual or non-numeric subjects
- Dependency on dataset quality and misconception accuracy

Technical Constraints:

- Designed for batch processing rather than real-time feedback
- Potential variability in LLM consistency across different runs
- Requires significant computational resources for large-scale deployment

Future Considerations:

- Need for extensive validation across diverse educational contexts
- Integration requirements for existing educational platforms
- Scalability challenges for institutional deployment

Future Work and Improvements

1

Technical Enhancements:

- Multimodal Support: Integration of mathematical diagrams and handwritten equations
- Cross-Domain Testing: Extension to science, language arts, and other subjects

2

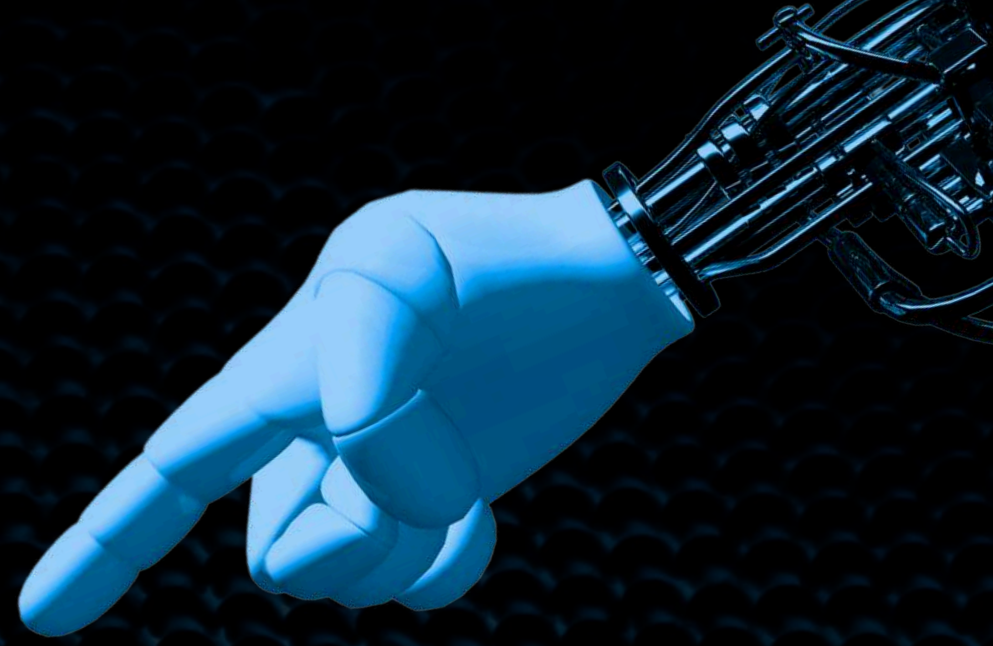
Educational Applications:

- Integration with existing Learning Management Systems
- Development of personalized remediation strategies

3

Research Directions:

- Comparative studies with human expert labeling
- Long-term impact assessment in classroom settings
- Advanced prompt engineering techniques for LLMs



Conclusions

We successfully developed an intelligent system for mathematics misconception detection achieving a MAP@25 score of 0.811. The system integrates local LLM and cybernetic feedback mechanisms, creating a scalable solution that automates educational assessment and reduces manual effort, establishing the foundation for future intelligent tutoring systems through a robust architecture that handles educational ambiguity.

Thank You