Francisco Jose De Caldas District University
Faculty of Engineering

Steven Navarro Parrales
20221020048

Juan David Amaya Patiño
20221020057

David Santiago Garcia Galeano
20231020158

Eng. Carlos Andrés Sierra
SYSTEMS ANALYSIS AND DESIGN
Group: 020-82

Bogotá
2025

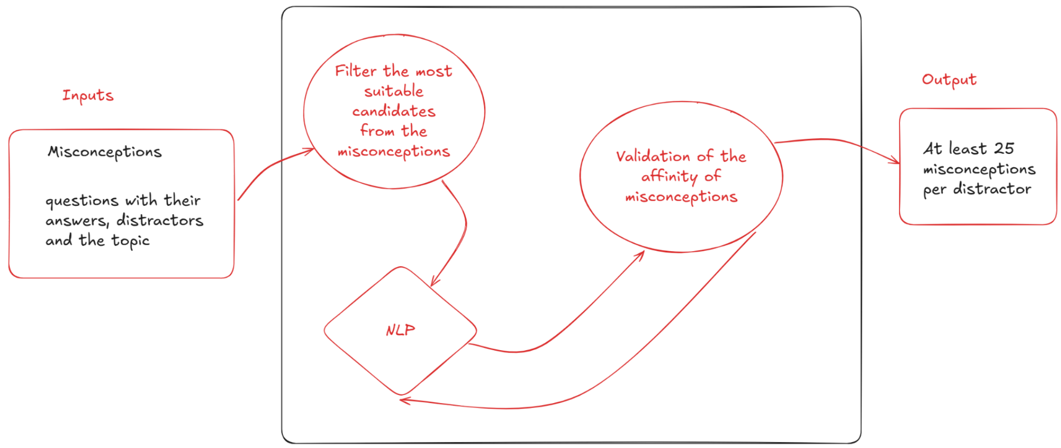
Competition link:
https://www.kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics/overview

Competition Overview: Predicting the affinity between misconceptions and incorrect answers (distractors) in multiple-choice questions.

teaching-learning processes through artificial intelligence. One of the main findings is the efficient automation of misconceptions labeling, which addresses an important pedagogical need and how metrics such as MAP@25 and computational constraints not only guide the development of effective solutions, but also applicable in real-world contexts. Together, this system represents an effective synergy between technology, education, and data science to address a high-impact educational problem.

System Design Document

1. Review Workshop #1 Findings:



The main features of the data are the misconceptions, which are concepts that help identify where students are making mistakes in multiple-choice questionnaires. These questions will be entered into the system along with the correct answer and the three distractor or trick options. The goal of the model is to label and generate misconceptions through training, linking the distractor choices with the concepts students are struggling with. Ultimately, the expected output is twenty-five misconceptions associated with each distractor.

The system includes restrictions to minimize unexpected errors as much as possible, by ensuring that all questions are within the field of Mathematics. This is because the quality of responses in this area can be measured objectively. Each question must have one correct answer and three distractors. Every question is linked to a specific topic to narrow down the search for possible misconceptions related to the distractors. A minimum of 25 misconceptions is required for each distractor. The associated misconceptions are then organized in descending order based on their level of affinity with the distractor.

The system for detecting misconceptions in mathematics is highly sensitive to small changes, such as modifying a question or removing a distractor, which can lead to significantly different results. This is due to the complexity of natural language and the model that interprets it, which reacts non-linearly to subtle variations. Additionally, subjectivity in data labeling and class imbalance further exacerbate this instability. To improve the system's stability, strategies such as using ensemble models, smoothing labels, and conducting controlled experiments to analyze the impact of changes on predictions are proposed.

2. **Define System Requirements:**

Functional Requirements

- **RF1 - Load questions**
  The system should allow for uploading multiple-choice questions, along with their correct answers, distractors, and the corresponding topic.

- **RF2 - Load misconceptions**
  The system should allow you to load a bank of misconceptions, each with its associated text and ID.

- **RF3 - Analyze distractors**
  The system must analyze each distractor to identify linguistic or conceptual patterns that may lead to misconceptions.

- **RF4 - Generate misconceptions**
  The system must generate at least 25 plausible misconceptions per distractor using a language model.

- **RF5 - Associate misconceptions with distractors**
  The system must label each misconception generated with the corresponding distractor.

- **RF6 - Export results**
  The system must allow exporting the generated misconceptions in a readable format such as CSC or Json.

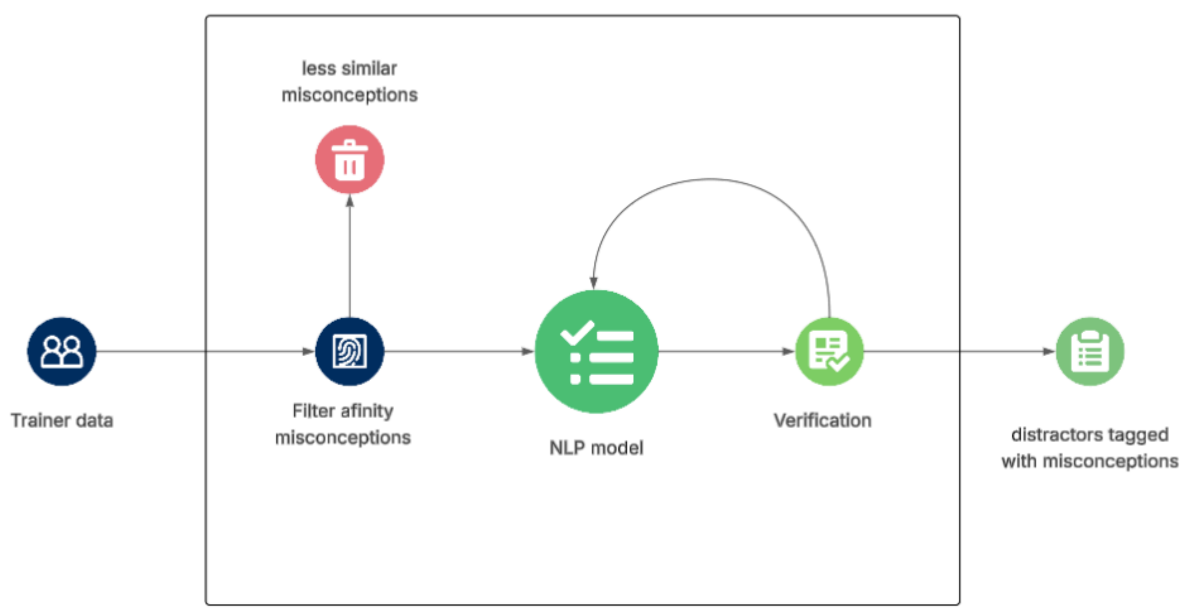
Non-Functional Requirements

- **RNF 1 - Response time**
  The system must generate the 25 misconceptions per distractor in the shortest possible time.

- **RNF2 - Scalability**
  The system must be able to process large batches of questions and large batches of concepts.

- **RNF4 - Integration of LLM models**
  The system must support at least one language model (such as GPT, LLaMA, etc.)

3. High-Level Architecture:



- **Trainer Data:** The training data consists of the question bank, including the corresponding text, the correct answer, and the distractors, as well as the misconceptions bank.

- **Filter Affinity:** The filtering of the most relevant misconceptions for each distractor is then carried out.

- **Less Similar:** All misconceptions with an affinity below a certain threshold are discarded.

- **NLP model:** The association of the misconceptions and their tagging by affinity is then carried out, ordering the misconceptions for each distractor from highest to lowest affinity.

- **Verification:** The association of distractors with their misconceptions is verified, and if this association does not meet a certain score, it is returned to the NLP model for re-training.

- **Distractor Tagged:** All distractors with their tagged misconceptions, ordered by affinity, that met the score from the verifier.

The system follows a defined life cycle, from the collection and structuring of data (Trainer Data), through the filtering stage of relevant misconceptions (Filter Affinity and Less Similar), to the automatic association of these with distractors using a natural language processing model (NLP model), followed by a quality check (Verification), and ending with the output of valid labeled distractors (Distractor Tagged).

An iterative mechanism is incorporated through automatic feedback from the verifier. If an association does not meet the required score, it is returned to the NLP model for retraining or fine-tuning, thus promoting continuous system improvement.
Complexity is managed in a modular manner, allowing for the development, testing, and optimization of each component separately. Decisions within the system are based on quantifiable data, such as semantic affinity and validation scores, which facilitates an objective evaluation of performance. Furthermore, overall system optimization is prioritized, focusing on ensuring that the final association between distractors and misconceptions has educational, not just technical, value. Finally, the system integrates

verification and validation mechanisms, ensuring both compliance with technical rules and the pedagogical relevance of the generated associations.

4. **Addressing Sensitivity and Chaos:**
The system minimizes sensitivity and avoids chaotic behavior by implementing a structured workflow that begins with filtering out disparate misconceptions. This reduces noise and improves the consistency of the training data. Because it works with natural language, it avoids and reduces semantic and syntactical errors in questions, distractors, and misconceptions. These parameters also prevent special characters from being filtered out, as these potential cases generate conflicts with the model, which can lead to poor learning or misdetection. The NLP model then processes only relevant information, reducing the possibility of erroneous interpretations caused by minor variations in the questions or distractors. Finally, a verification stage validates predictions before labeling distractors, introducing an additional layer of control. This cascading approach reduces the impact of unstable initial conditions and mitigates systematic errors, stabilizing system performance.

5. **Technical Stack and Implementation Sketch:**
We consider that the tools that could serve us for the implementation of the system are python as a programming language since it is very flexible and has a lot of support in the subject of machine learning models, also for the part of natural language processing (NLP) we chose to use the Anthropic library, to use its api and access its Claude models, and use the Claude 3.5 Sonnet model for its high quality of reasoning and text comprehension, finally we will use an llm like GPT4 for the validation of the labels since the classification of concepts is usually not very understandable for the models, then in this way we will reduce the margin of error, and in the event that sufficient affinity of the misconceptions is not approved, the NLP will be retrained to correct errors.