Built RAG with Open Source

Ricardo Jose Molina Gonzalez presents a solution for document misclassification using open-source RAG. The presentation covers document classification, indexing, and question answering. It also details experiments with different LLM models.

Ricardo José Molina González

Student of Applied Artificial Intelligence at Miami Dade College. Originally from Nicaragua, focused on innovative solutions with open source code.

1

Passion for AI

Dedicated to exploring the potential of artificial intelligence.

2

Practical Approach

Implementing real solutions for concrete problems.

Agenda

1. Issue Solve: Loans Documents
2. Classification: Before and After
3. Experiments: LLM Models - Phi-2
4. Experiments: LLM Models - Mistral
5. Experiments: Sentence Transformers
6. Performance & Improvements
7. Future Steps
8. Live Demo

Issue Solve: Loans Documents

The core problem is the misclassification of various loan documents, often due to differing formats. Our solution addresses this by extracting data from diverse PDF documents, accurately classifying unknown document types, and then indexing them with meaningful labels. This enables the system to effectively answer questions based on the document content.

Classification: Before and After

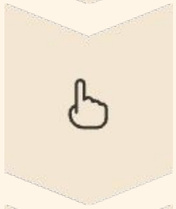
The diagram illustrates the RAG process.

Ingest PDF Documents

Load diverse PDF documents into the system.

Automatic Classification

Automatically categorize documents.



Index Documents

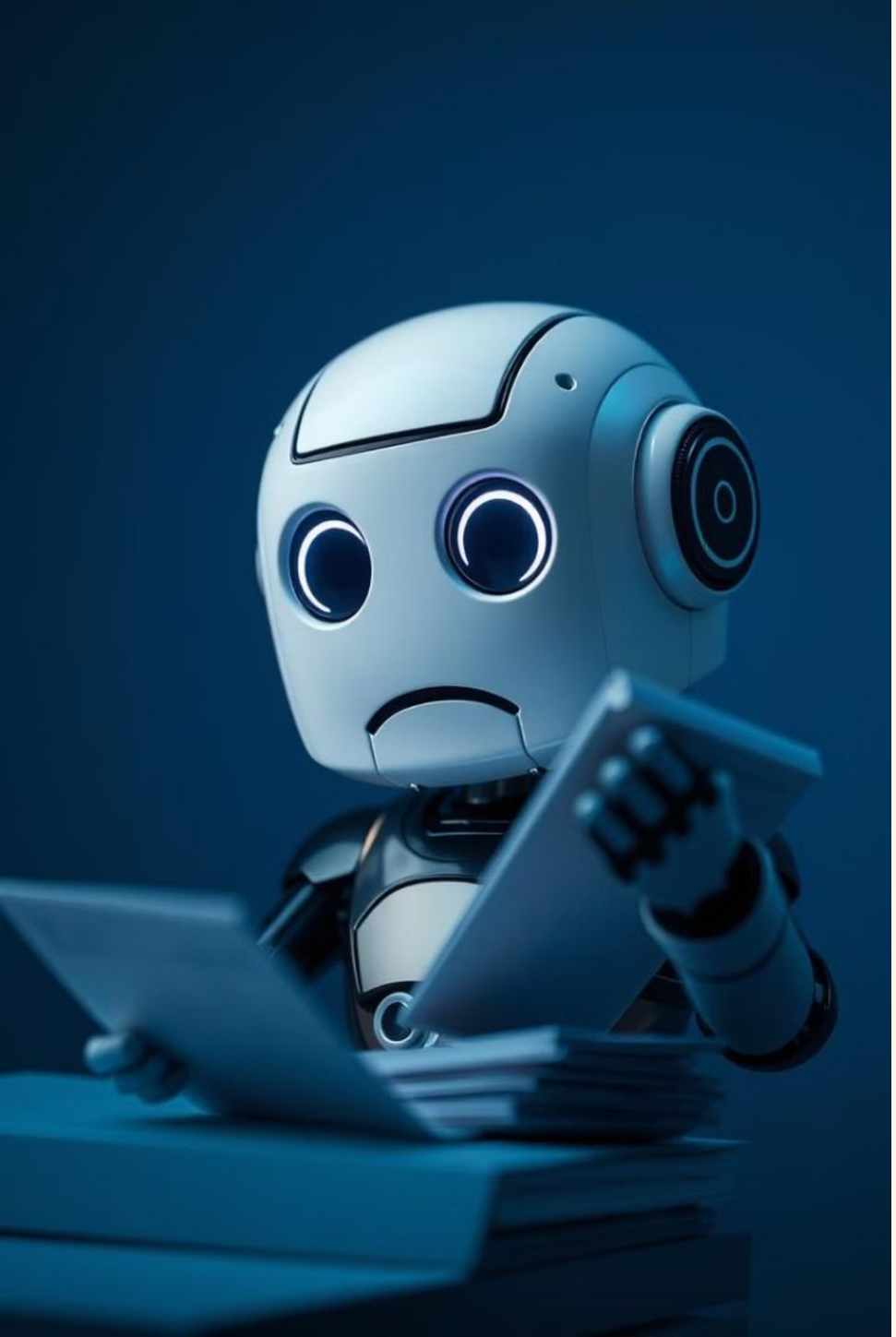
Index classified documents for efficient retrieval.

Synthesize Answer Retrieval

Retrieve relevant information for user questions.

Answer Questions

Provide answers based on document information.



Experiments: LLM Models - Phi-2

Phi-2 exhibited challenges in document classification due to its smaller parameter size (2.7 billion). This resulted in increased development time and lower performance in sentiment analysis compared to other models.

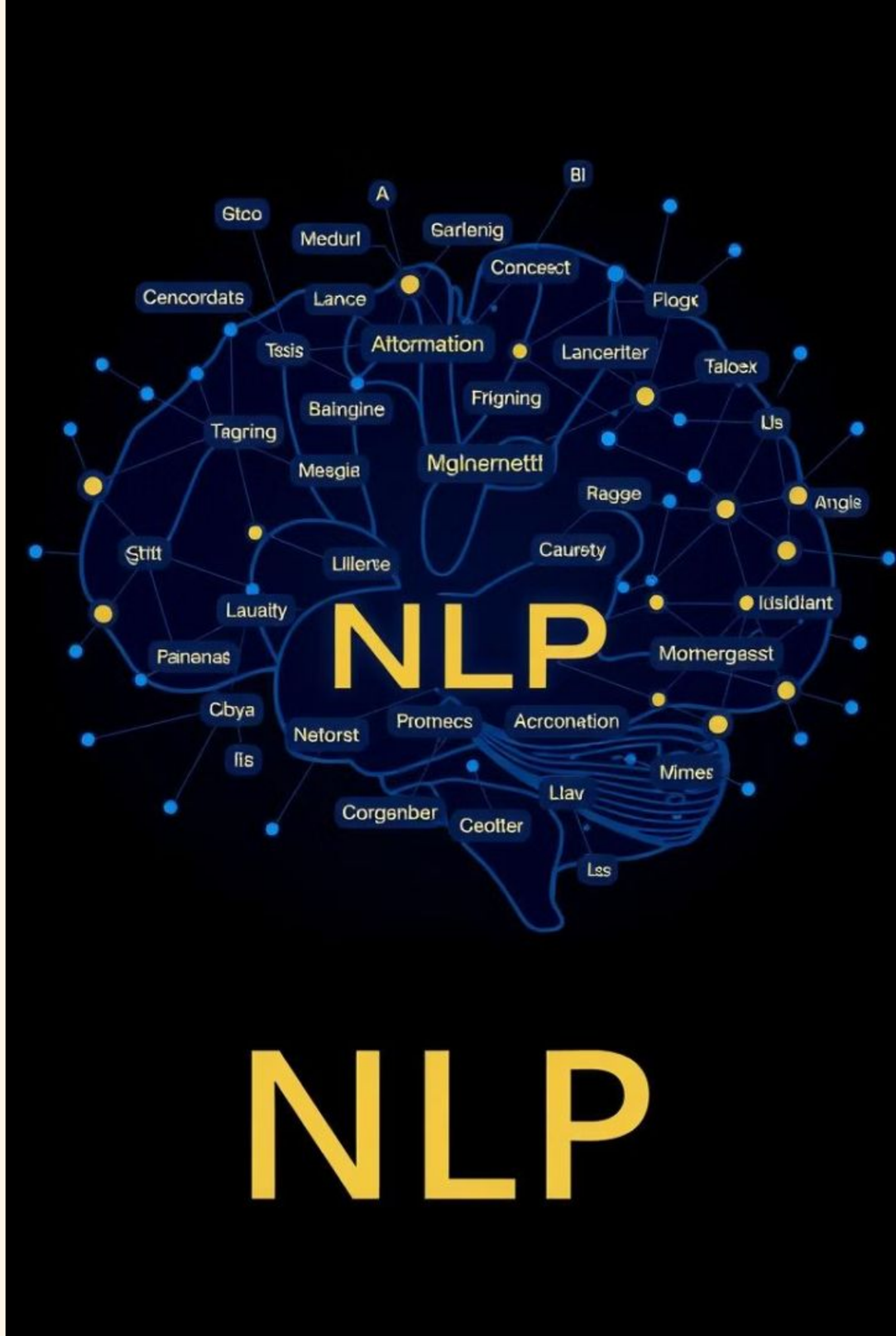
Experiments: LLM Models - Mistral

Mistral classifies documents more effectively. It has more parameters (7 Billion). It reduces the development time by half. It also has more accuracy in sentiment analysis.

Experiments: Sentence Transformers

sentence-transformers/all-MiniLM-L 6-v2, with 22 million parameters and 6 transformer layers, results in increased processing time and lower accuracy in semantic tasks.

sentence-transformers/all-mpnet-bas-e-v2, featuring 110 million parameters and 12 transformer layers, improves processing time and semantic task accuracy.



Performance & Improvements

- Enhanced Classification Accuracy: Achieved correct classification of diverse PDF documents, significantly reducing misclassification errors.
- Improved Semantic Embeddings: The refined embedding model delivers enhanced semantic classification, ensuring more relevant document indexing.
- Context-Aware Prompting: Detailed prompts are now effectively utilized to elicit accurate and comprehensive responses from the system.

Future Steps

- Indexing by Clients: Enable clients to index all necessary documents.
- Improve the Prompt: Refine prompt modeling to allow the system to retrieve accurate answers.
- Better Model Analysis: Conduct more experiments to obtain improved statistical performance metrics.
- Identify Duplicated and Incorrect Data: Enable the system to isolate incorrect or duplicated data for review.