Automated Abstract Tagging: Enhancing Peer-Reviewed Abstract Categorization with MeSH Ontology and Large Language Models

Michael Long
Kate Burdekin
Chris Coxen
Demian Pasquarelli

Agenda

- Background
- Objective
- Methodology
- Results and Comparison
- Benefits and Implications
- Future Work
- Conclusion

Background

Project Background

Problem

- The peer-review process for government-funded applications requires significant resources:
  - Large investment of time from Subject Matter Experts (SMEs)
  - Extensive administrative tasks
  - Involvement of NIH leadership
- Manual reading and categorization of abstracts is time-consuming and labor-intensive
- Challenge in efficiently matching proposals to appropriate reviewers

Solution

- Automated abstract tagging system leveraging:
  - MeSH (Medical Subject Headings) ontology
  - Large Language Models (LLMs)
- Key features:
  - Locally installed, secure, and open-source LLM
  - Automatic identification of MeSH terms from abstract text
  - No risk of data transmission to the cloud

What Are MeSH Terms?

- Developed by the National Library of Medicine
- Comprehensive controlled vocabulary for the biomedical domain
- Key features:
  - Hierarchically-organized: Terms are arranged in a tree-like structure
  - Controlled: Ensures consistency in indexing and searching
  - Regularly updated: Reflects current biomedical terminology
- Used for:
  - Indexing articles in MEDLINE/PubMed
  - Cataloging books and other materials
  - Facilitating precise searches in biomedical literature
- Benefits:
  - Standardizes terminology across the field
  - Enables more accurate and efficient literature searches
  - Supports automated categorization of biomedical texts

What Are Large Language Models

- Advanced AI systems trained on vast amounts of text data
- Capable of understanding and generating human-like text
- Key characteristics:
  - Deep learning-based: Utilize neural networks with many layers
  - Context-aware: Can understand and maintain context in text
  - Versatile: Applicable to various natural language processing tasks
- Relevant capabilities for our project:
  - Text comprehension: Can "read" and understand abstract content
  - Named Entity Recognition: Can identify specific terms* (like MeSH terms) in text
    - * Kind of

What Are Large Language Models

- Advantages for abstract tagging:
  - Can process large volumes of text quickly
  - Capable of “understanding” complex biomedical terminology
  - Can be implemented in various ways to suit specific needs
- Considerations:
  - Need for secure, locally-installed versions to protect sensitive data
  - Importance of accuracy and reliability in identifying relevant terms
  - Potential for integration with existing systems and vocabularies

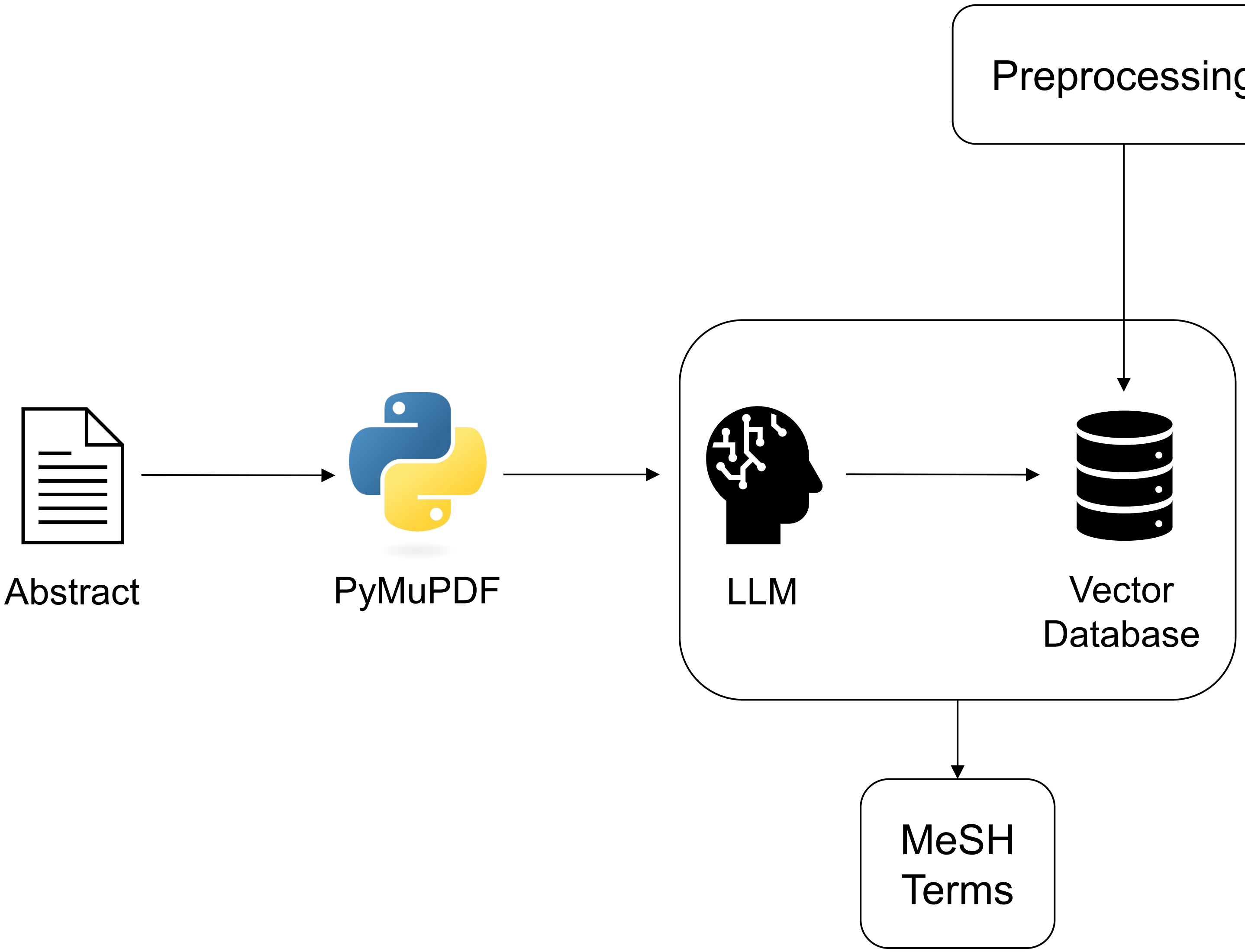
Objective

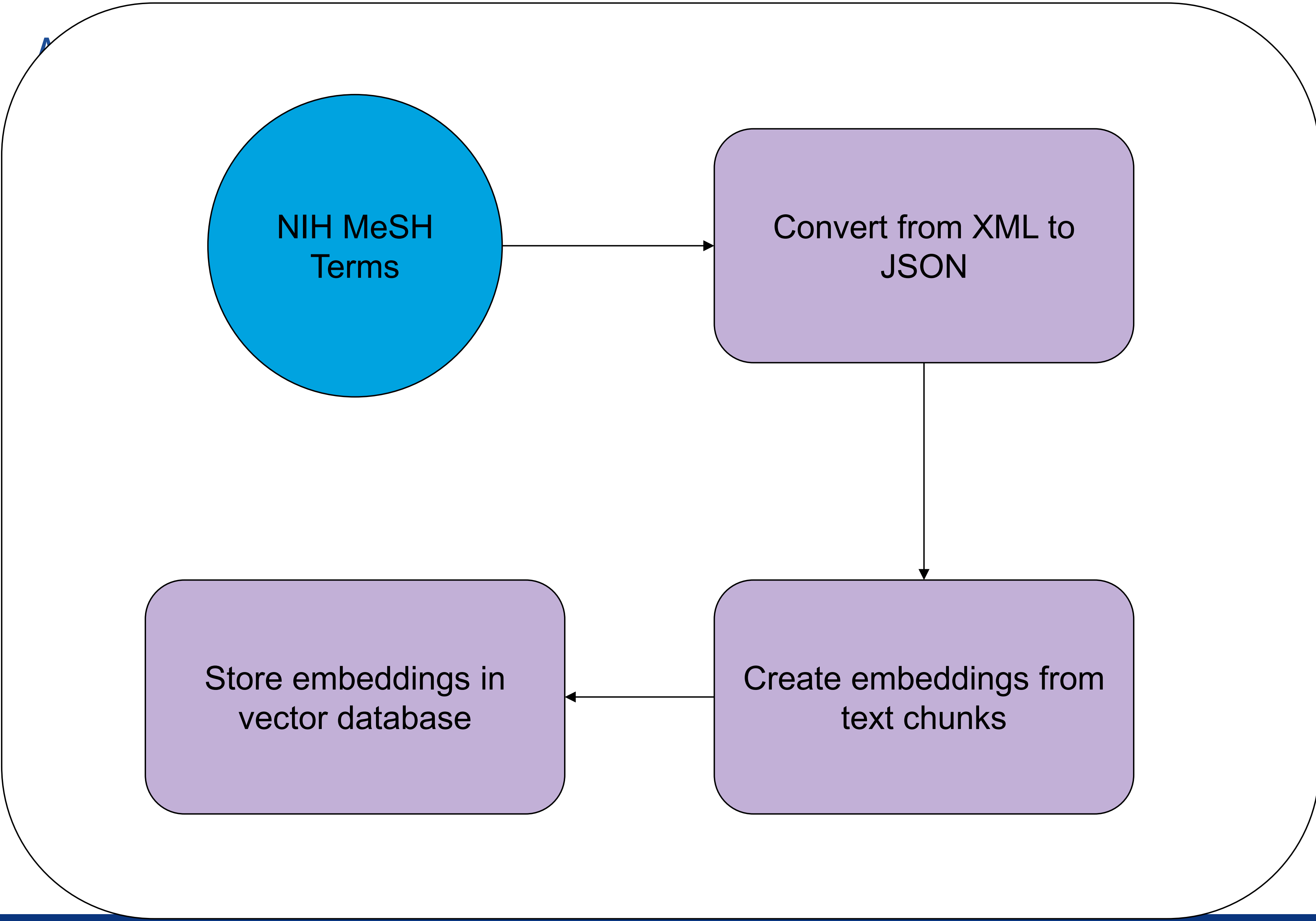
Objective

- Our primary objective was to develop an automated system for tagging peer-reviewed abstracts using MeSH ontology, leveraging the capabilities of LLMs.
- Key aims:
  - Automate the process of reading and categorizing scientific abstracts
  - Reduce the manual effort required from SMEs and administrative staff
  - Improve the efficiency and accuracy of abstract categorization
  - Facilitate better matching of proposals to appropriate reviewers
  - Enhance the overall peer-review process for government-funded applications
- By achieving these aims, we sought to streamline the peer-review process, allowing SMEs to focus more on evaluating the merit of applications rather than on time-consuming administrative tasks.

Methodology

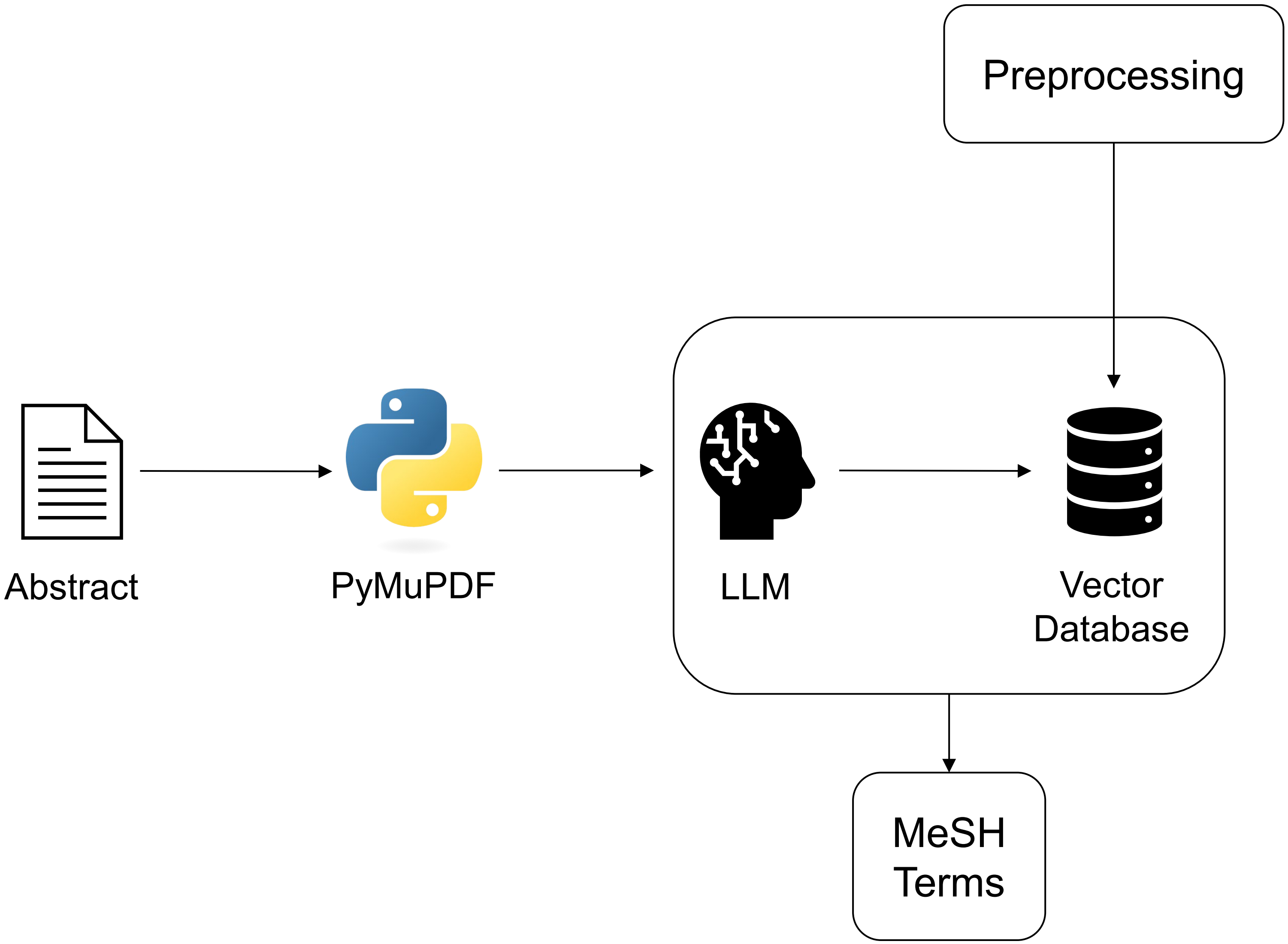
Architecture

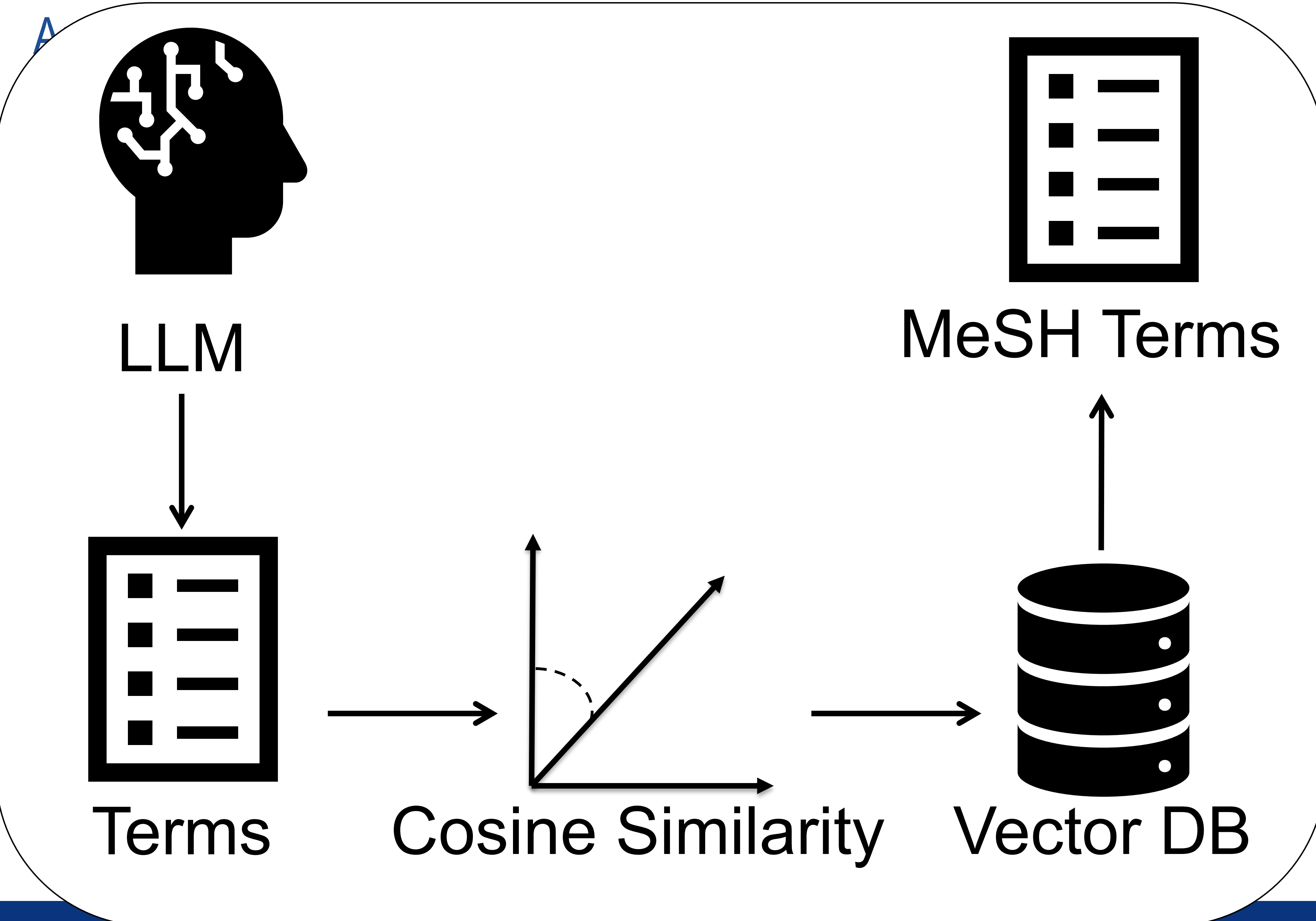



NIH MeSH Terms
Convert from XML to JSON
Create embeddings from text chunks
Store embeddings in vector database

Preprocessing

Architecture




LLM
MeSH Terms
Terms
Cosine Similarity
Vector DB

Results and Comparison

Comparing Results

- Compared between:
  - LLM
  - Human tagger
  - NIH online tool
- Used Jaccard Similarity
  - Number of observations in both sets / number of overlapping observations in both sets
  - Heavily penalizes datasets that have a large difference in the total number of observations
- Sample sentences from five different abstracts across four domains:
  - AI Readiness
  - Cloud
  - Data Repo
  - Software
- Running the full extraction process (text extraction from PDF, run the LLM, get results from database) took ~34 minutes
  - 5 models * 151 abstracts = ~2.7 seconds per prompt

Comparing to the NIH Online Tool

Domain	Model Average	Human Review Average
AI Readiness	**0.1724**	0.0678
Cloud	**0.2011**	0.0247
Data Repo	**0.1816**	0
Software	**0.2053**	0.0316

Model Results

- These results are from comparing the models to the NIH online tool
- When looking at the overall averages across the 20 abstracts, the Llama 3.1 8b model performed best overall
- When looking at the averages within the abstract domains:
  - AI Readiness: Mistral-Nemo 12b
  - Cloud: Llama 3.1 8b
  - Data Repo: Llama 3.1 8b
  - Software: Llama 3.1 8b

Benefits and Implication

Immediate Benefits

- Time and resource savings
  - Significant reduction in manual tagging time
  - Freeing up SME resources for more critical tasks
- Improved consistency
  - More standardized approach to abstract categorization
- Scalability
  - Ability to handle large volumes of abstracts efficiently
  - Potential for application across multiple research domains
- Enhanced security
  - Local processing ensures data privacy and security

Long-term Implications

- Streamlined peer review process
  - Faster initial categorization of research proposals
  - More efficient allocation of proposals to appropriate reviewers
- Improved research discovery
  - More consistent tagging could enhance searchability of research
  - Potential for better cross-disciplinary connections
- Data-driven decision making
  - Aggregated tagging data could provide insights into research trends
  - Potential to inform strategic funding decisions
- Extensibility to other tasks
  - Possibility of extending the system to reviewer-proposal matching
- Continuous improvement
  - As the system processes more abstracts, there's potential for ongoing refinement and increased accuracy

Future Work

Future Work

- Implement a full RAG pipeline
  - Enhance embeddings with LLM-generate narrative for each MeSH term
  - Include a re-ranker for embeddings
- Investigate specialized embedding models
- Implement human-in-the-loop
  - Store in traditional database with a web interface for verification and editing
- Expand scope to include tagging for reviewers
  - This would allow us to match reviewers to domain-relevant abstracts

Conclusion

Conclusion

- Innovative Approach
  - Successfully combined MeSH ontology with LLMs for automated abstract tagging
  - Implemented a "pseudo-RAG" approach for accurate term matching
- Improved Efficiency
  - Significantly reduced time and effort required for abstract categorization
  - Demonstrated higher consistency compared to manual tagging
- Competitive Performance
  - Achieved higher similarity to NIH tool compared to human taggers
  - Maintained data security through local processing
- Future Potential
  - Opportunities for further improvement and expansion of the system
  - Possible applications in reviewer matching

Thank you

Contact: Michael Long | email: michaellong@rti.org

Appendix A: Technology Used

- Python
- Ollama for interfacing with models
  - Uses 4-bit quantized versions
- LanceDB for vector database
- sentence-transformers library
  - "all-MiniLM-L12-v2"

Appendix B: Model-Tool Results

	Model					
Abstract Domain	**Gemma 2 9b**	**Llama 3.1 8b**	**Llama 3.2 3b**	**Mistral-Nemo 12b**	**Qwen 2.5 7b**	**Average**
AI Readiness	0.1580	0.2229	0.1185	**0.2232**	0.1392	0.1724
Cloud	0.1999	**0.2379**	0.1849	0.2201	0.1628	0.2011
Data Repo	0.1625	**0.2219**	0.1696	0.1642	0.1899	0.1816
Software	0.2047	**0.2236**	0.2209	0.1647	0.2127	**0.2053**

Appendix C: Model-Human Results

	Model					
Abstract Domain	**Gemma 2 9b**	**Llama 3.1 8b**	**Llama 3.2 3b**	**Mistral-Nemo 12b**	**Qwen 2.5 7b**	**Average**
AI Readiness	0.0435	0.0327	0.0279	**0.0606**	0.0557	0.0441
Cloud	**0.0625**	0.0340	0.0478	0.0404	0.0384	**0.0446**
Data Repo	0.0222	**0.0286**	0.0125	0	0.0258	0.0178
Software	0.0404	0.0338	0.0386	**0.0582**	0.0243	0.0390