

REPORT GENIE

CSCE 670 – Information Storage and Retrieval Initial Project Proposal

Afreen Ahmed
535005867

Rhea Sudheer
936002058

Aniruddh Adiga
236002123

Rushil Ravi
836000314

What is exactly the function of your tool? That is, what will it do?



Report Genie is a tool we are designing to help users better understand their medical reports. All they would need to do is upload their medical reports and Report Genie will use a Retrieval-Augmented Generation model to query and receive explanations, diagnoses, and relevant insights from the report.

Why would we need such a tool and who would you expect to use it and benefit from it?

As patients, making informed decisions about our health is essential for receiving the best care and improving our well-being. Medical reports can be difficult to understand, but having the right tools can help us interpret diagnoses and treatment plans. When we understand our health conditions, we can ask better questions, follow recommendations more effectively, and take an active role in our care. Report Genie can be useful for patients, caregivers and healthcare providers.

Does this kind of tool already exist? If similar tools exist, how is your tool different from them? Would people care about the difference? How hard is it to build such a tool? What is the challenge?

Some tools exist to explain medical terms (like MedlinePlus), but they typically don't offer a full, interactive querying system specific to individual reports. ReportGenie is different because it offers personalized results by allowing users to directly upload their own medical reports. This enables the tool to analyze the specific content of each report and provide tailored explanations, insights, and answers based on individual health data.

How do you plan to build it? You should mention the data you will use and the core algorithm that you will implement.

We will develop a custom RAG system that extracts relevant medical information from reports and existing medical knowledge base and generates user-friendly explanations in layman terms.

- **Data Collection:** Gather clinical text from open-source datasets:
 - [PubMed Open Access](#) – Biomedical literature for relevant medical research.

- [MTSamples](#) – A dataset of transcribed medical reports for training models.
- [LOINC](#) – Standardized lab test codes and names for mapping blood test results.
- [SNOMED CT](#) – Medical terminology system for lab test explanations.
- [Harrison's Principles of Internal Medicine-2 volume set](#)
- **Retrieval System:** Build a custom FAISS-based search engine for medical knowledge extraction.
- **Generation Pipeline:** Use retrieved text and generate explanations using the existing NLP model.

What existing resources can you use?

<https://github.com/SimoneParvizi/Medical-Information-retrieval-with-RAG>

<https://github.com/chaoyi-wu/PMC-LLaMA>

How will you demonstrate the usefulness of your tool?

- **Basic User Testing:** Have a small group of users upload sample medical reports and ask questions. Check if they can understand their blood test results better with the tool's explanations.
- **Performance Metrics:** Manually verify if our FAISS-based retrieval system is fetching the right medical terms and relevant explanations. Evaluate how accurately the tool can answer medical queries by comparing it with domain experts or gold standard answers.
- **Error Analysis:** Identify cases where the tool provides incorrect or unclear answers and refine the retrieval process accordingly.
- **Basic Benchmarking:** Use **BioASQ datasets** (medical question-answering benchmarks) to check how well the tool responds to standard medical queries.

A rough timeline to show when you expect to finish what. List a couple of milestones.

| Milestones | Description |
|------------|---|
| Weeks 1-3 | Gather data and preprocess medical reports. Set up a dev environment and define core features. Implement the retrieval system (use embeddings for information retrieval) and train the RAG model on medical data. |
| Weeks 3-7 | Integrate generation capabilities for answering queries and providing explanations. Build the user interface (web app), implement security features, and test the tool with initial users. |
| Week 8 | Conduct final testing, refine the tool based on feedback, and prepare a demonstration and report. |