Abstract Reliable biomedical question answering is a critical challenge as Large Language Models (LLMs) often generate hallucinated or unsupported statements. Med-RAPTOR Agent is a hybrid Generative + Agentic AI architecture designed to produce evidence■backed, verifiable, and trustworthy medical answers. The system integrates RAPTOR■style hierarchical retrieval with an NLI■driven verify■and■revise loop to ensure factual consistency. Retrieved biomedical literature such as PubMed abstracts is organized into a multi■level retrieval index, enabling efficient and context■aware grounding. The agent generates answers, validates each claim using medical NLI models (MedNLI, SciNLI), and regenerates unsupported parts, ensuring citation■based reliability. The project is expected to demonstrate reduced hallucinations, improved interpretability, and a transparent evidence pipeline compared across three systems: Vanilla RAG, RAPTOR hierarchical retrieval, and the full Agentic RAPTOR. This work has significant implications for building safe and trustworthy medical AI systems. **Introduction** Biomedical decision■making requires precise, verifiable, and evidence■grounded information. However, state■of■the■art LLMs frequently hallucinate or produce unverifiable medical statements, making them unreliable for clinical use. As medical information is high■stakes, even minor inaccuracies can mislead clinicians and patients. Existing Retrieval■Augmented Generation (RAG) reduces hallucination but still struggles with shallow retrieval, lack of hierarchical structure, and no intrinsic verification of generated content. To address these limitations, we propose Med■RAPTOR Agent, a hybrid generative + agentic system designed to elevate reliability in biomedical QA. The system combines: (1) RAPTOR■style hierarchical document indexing for multi■level retrieval, (2) generative summarization for structured evidence use, and (3) an NLI■based verify■and■revise loop that enforces factual consistency. Through this integration, every generated sentence is checked against PubMed■style evidence, and only supported claims are presented. Our objective is to create a transparent, interpretable, and verifiable pipeline that outperforms standard RAG systems. The project demonstrates how generative AI can be made trustworthy in sensitive domains such as medicine. **Methodology** 1. Data Collection & Pre■processing We use biomedical datasets such as PubMedQA, BioASQ, MedQA, and PubMed abstracts. The raw text is cleaned, normalized, chunked, and embedded using biomedical embedding models like Sentence■BERT or PubMedBERT. All documents are stored in vector format for retrieval. 2. RAPTOR Hierarchical Indexing RAPTOR constructs a hierarchical tree of summaries. First, biomedical passages are embedded and clustered. Each cluster is summarized using a generative model, creating mid■level summaries. Clusters of clusters form higher■level summaries until a multi■level retrieval tree is formed. This allows retrieval at varying abstraction levels, improving recall and contextual coherence. 3. Baseline RAG Pipeline We build a vanilla RAG model using FAISS for similarity search. For each medical query, top■k context documents are retrieved and provided to an LLM such as BioGPT, Llama■3■Instruct, or a similar generative model. The system produces an initial answer grounded in retrieved evidence. 4. Agentic Verify■and■Revise Loop This is the key innovation of our system. Each generated sentence is validated using an NLI model (MedNLI, SciNLI, or DeBERTa). If the evidence contradicts or does not support the claim, the agent triggers a revision cycle. The LLM regenerates only unsupported parts or rewrites the passage using stronger evidence. This loop continues until every statement is fully supported. 5. Evaluation We evaluate using Exact Match (EM), F1 scores for supervised QA tasks, and qualitative evaluation for open■ended medical queries. FactScore and citation accuracy metrics measure evidence grounding. We compare: - Vanilla RAG - RAPTOR hierarchical retrieval - Full Agentic RAPTOR pipeline We also visualize the retrieval hierarchy and agent reasoning traces. 6. Tools & Frameworks The project uses PyTorch, HuggingFace Transformers, FAISS, scikit■learn, LangChain/LlamaIndex, and Matplotlib. Embedding models include SBERT and PubMedBERT. NLI models include MedNLI, SciNLI, and DeBERTa. The system is designed to be modular and extensible. **Individual Member Contribution** Archie Gaur (SR 24826) - Implement RAPTOR hierarchical indexing - Build FAISS vector database - Lead dataset preparation and embedding pipeline - Assist in RAG baseline development Arpit Mishra - Implement NLI■driven verify■and■revise agentic loop - Integrate MedNLI/SciNLI with the generative pipeline - Design evaluation framework and automated scoring - Optimize agentic reasoning workflow Sreeja - Develop system architecture, diagrams, and pipeline documentation - Implement retrieval tree visualization tools - Assist in RAPTOR summarization and LLM integration - Prepare experimental comparison charts Rahul Dewangan - Work on final integration of RAG, RAPTOR, and Agentic RAPTOR - Conduct testing, debugging, and model performance tuning - Contribute to final report writing and presentation preparation