DSCI 6004: Natural Language Processing

Term Project: RAG System Development and LLM Comparison

Proposal Slides (5%) Due Friday 03/21/2025 11:59pm ET

Code Demo and Presentation (10%) Due Friday 04/25/2025 11:59pm ET

Final Submission (15%) Due May 05/02/2025 11:59pm ET

Project Overview

The purpose of this term project is to demonstrate your practical skills in applying Retrieval-Augmented Generation (RAG) systems and comparing the performance of different Large Language Models (LLMs). Students may do final projects solo, or in teams of up to 3 people. We strongly recommend you do the final project in a team. Larger teams are expected to do correspondingly larger projects, and you should only form a 3-person team if you are planning to do an ambitious project where every team member will have a significant contribution.

Project Requirements

For this term project, you will:

1. Develop a Retrieval-Augmented Generation (RAG) system on a topic of your choice
2. Implement your RAG system using three different free or open-source LLMs (e.g., Llama-3, Mistral-7B, Phi-3, RWKV, Falcon, etc.)
3. Create at least 10 domain-specific questions about your chosen topic
4. Evaluate and compare the responses generated by each LLM
5. Analyze the differences in performance, accuracy, and approach between the models

Milestones

First Milestone: Project Proposal

Deadline: Friday 03/21/2025 11:59pm ET

Prepare and submit a project proposal consisting of a maximum of 9 slides, outlining the following:

- Team Members (on the cover page)
- Project topic and the domain/corpus you will use for your RAG system
- Statement of project objectives
- Statement of value - why is this project worth doing?

- Review of the state of the art in RAG systems and relevant works (max. 2 slides - include citations)
- Approach (which free/open-source LLMs you plan to use, what datasets/knowledge base you'll incorporate, and your system architecture)
- Deliverables (i.e., a list of items that will be submitted upon completion, and their relevance to the stated objectives)
- Evaluation methodology (including metrics for comparing LLM outputs and RAG system effectiveness)

Second Milestone: Code Demo and Presentation

Deadline: Friday 04/25/2025 11:59pm ET

Record a short (< 10 minutes) demonstration of your project (i.e., running code and presenting your slides) to your peers. This video must be uploaded on YouTube, but if you wish to keep it private, you can choose to post it as an "unlisted" video.

Third Milestone: Final Submission

Deadline: May 05/02/2025 11:59pm ET

Submit a project report, composed as a paper written in the style of an ACL/NeurIPS/AAAI, etc. conference submission. It should include:

- An abstract and introduction
- Clear description of your RAG system and implementation
- The list of at least 10 domain-specific questions you developed
- Technical details of each LLM implementation
- Comparative results across the three LLMs
- Analysis of differences in response quality, factual accuracy, reasoning, and other relevant dimensions
- Discussion of strengths and weaknesses of each model
- Proper citations throughout (you'll probably want to cite at least 5-10 papers)

If you are working in a team of two or three, the paper should be on the order of 8 pages excluding references; working alone, you should target more like 5-6 pages. Don't treat these as hard page requirements or limits and let the project drive things.

Upload your completed project (including code and slides) to a new Github repository, along with a user documentation manual as a .MD file describing the project and usage instructions to other interested students and researchers.

Grading Rubric

We will grade the project reports according to the following rubric:

Clarity/Writing (3 points): Your paper should clearly convey your RAG system architecture, the LLMs you employed, and your comparative analysis methodology.

Implementation/Soundness (5 points): Is the RAG implementation technically sound? Do you describe what seems like a convincing implementation? Is the experimental design for comparing LLMs appropriate?

Results/Analysis (7 points): Provide thorough analysis of how each LLM performed on your questions. Identify patterns in their responses, strengths and weaknesses, factual accuracy, and reasoning capabilities. Whether the results favor one model or show mixed performance, try to provide examples and analysis. Discuss why certain models might perform better on particular types of questions.

Suggestions for RAG System Development

Your RAG system should include something novel in either the retrieval mechanism, the knowledge base construction, the prompt engineering, or the evaluation methodology. Your end goal shouldn't be just implementing a basic RAG system, but rather exploring interesting dimensions of RAG performance across different LLMs.

Some potential areas to focus on include:

- Domain-specific knowledge bases (e.g., legal, medical, scientific, financial)
- Comparison of different retrieval mechanisms (vector search, hybrid search, etc.)
- Evaluation of hallucination rates across different open-source LLMs in a RAG context
- Analysis of how different open-source LLMs handle ambiguity or uncertainty in retrieved information
- Low-resource or multilingual RAG implementations
- Optimization techniques for running larger models on limited hardware
- Strategies to improve performance of smaller open-source models through prompt engineering
- Efficient RAG architectures suitable for resource-constrained environments

Be bold in your choice! This project is not graded solely on how well your system works, as long as you can convincingly show that your implementation is doing something meaningful, and your analysis is insightful.

Computational Resources and Model Selection

For this project, students are required to use only free or open-source LLMs to ensure accessibility for all. Some recommended options include:

1. **Locally runnable models**:
    - Llama-3-8B or similar smaller variants
    - Mistral-7B open-source models

    - Phi-3-mini or other smaller Microsoft models
    - Falcon models
    - RWKV models
2. **Free API access** (with reasonable usage limits):
    - Hugging Face's free inference API for open-source models
    - Ollama for local deployment
    - Google Colab's built-in access to some open models

The operating assumption is that Google Colab and your personal computers should be sufficient for running smaller models locally or accessing them through free APIs with reasonable latency. For larger models, consider using quantized versions (4-bit or 8-bit) that can run on consumer hardware.