10 Academy Cohort A

Weekly Challenge: Week 11

Contract Advisor RAG: Towards Building A High-Precision Legal Expert LLM APP

Business objective

Lizzy AI is an early-stage Israeli startup, developing the next-generation contract AI (see short video). We leverage Hybrid LLM technology (edge, private cloud and LLM services), to build the first, fully autonomous, artificial contract lawyer. The first step in our journey is a powerful contract assistant, with the ultimate goal of developing a fully autonomous contract bot, capable of drafting, reviewing, and negotiating contracts, independently, end-to-end, without human assistance.

Our task is as follows: **build, evaluate and improve** a RAG system for Contract Q&A (chatting with a contract and asking questions about the contract).

Background Context

What is RAG: Retrieval Augmented Generation, commonly known as RAG, is a hybrid AI model that marries the expertise of powerful language models with the richness of external data sources. At its core, RAG leverages a large language model for generating responses, but with a twist – it first retrieves relevant information from a vast pool of external data. This retrieval phase empowers the model to augment its generated responses with information that goes beyond its initial training data, offering more accurate, informed, and context-rich outputs.

Why is RAG Exciting: Building a basic RAG system can be surprisingly straightforward, making it an enticing entry point for AI enthusiasts and students. However, crafting a high-quality, robust RAG system that performs exceptionally well is a complex and challenging endeavor. This complexity provides a rich learning ground for those aspiring to push the boundaries in AI.

RAG Opportunities: As AI continues to evolve, the demand for models that can intelligently and dynamically interact with vast pools of information is skyrocketing. RAG sits at this juncture, proving its value in numerous applications - from enhancing search engines to powering sophisticated chatbots and decision-support systems. The industry is on a constant lookout for professionals who can innovate and improve RAG systems. Given its vast potential and applicability, experts in RAG are poised to be highly sought after in the AI industry. The ability to design, implement, and refine RAG systems aligns with the industry's need for AI solutions that are both knowledgeable and contextually aware, making RAG expertise a valuable and future-proof skill.

Please watch/read the following links as a background knowledge before starting with the task.

Key:

- Langchain for LLM Applications (video course).
    - (pay special attention to the chapters on Q&A and Evaluation).
- Advanced Retrieval for AI with Chroma (video course).
- RAGAS Evaluation with Langchain (blog post).
- 9 Effective Techniques to boost RAG performance (blog post).

Optional:

- Langchain RAG Evaluation Webinar (Webinar).
- RAG Overview and Advanced Techniques (Paper).
- Query Expansion by Prompting LLM (paper).
- Prompt Engineering Video Course (video course).
- OpenAI Prompt Engineering Guide (Guide).

Data for Evaluation

- Evaluation set which contains two Contracts (a short one and a long one) with a list of 10 questions and correct answers for each.

Learning Outcomes

Skills

- Advanced NLP Techniques: Proficiency in applying natural language processing to specialized domains.
- Machine Learning and AI Application: Skills in implementing, fine-tuning, and evaluating AI models.
- Data Science Proficiency: Expertise in data collection, curation, preprocessing, and analysis.
- Software Engineering: Enhanced abilities in API development, database management, and system integration.
- Problem-Solving and Critical Thinking: Improved problem identification and solution development skills.
- Performance Optimization: Skills in optimizing system performance and efficiency.
- Project Management: Experience in managing and executing complex AI projects.
- Communication and Collaboration: Strengthened team collaboration and project communication skills.

Knowledge

- Domain-Specific Legal Knowledge: Understanding of legal terminology and contract law nuances.
- AI and RAG Systems Trends: Insights into the latest advancements and trends in AI and RAG systems.
- Ethical AI and Legal Compliance: Awareness of ethical considerations and legal compliance in AI applications.
- System Evaluation Techniques: Knowledge in setting up and executing system evaluation metrics and benchmarks.
- Bias Detection and Mitigation: Understanding of bias in AI systems and techniques to mitigate it.

Team

Tutors:

- Yabebal
- Emtinan
- Rehmet

Key Dates

- **Discussion on the case** - 9:30 UTC time on Monday 19 February 2024. Use #all-week-11 to ask questions.
- **Interim Submission** -8:00 PM UTC time on Wednesday 21 February 2024.
- **Final Submission** - 8:00 PM UTC time on Saturday 24 February 2024

Badges

Each week, one user will be awarded one of the badges below for the best performance in the category below.

In addition to being the badge holder for that badge, each badge winner will get +20 points to the overall score.

> **Visualization** - quality of visualizations, understandability, skimmability, choice of visualization
> **Quality of code** - reliability, maintainability, efficiency, commenting - in future this will be CICD/CML
> **Innovative approach to analysis** -using latest algorithms, adding in research paper content and other innovative approaches
> **Writing and presentation** - clarity of written outputs, clarity of slides, overall production value
> **Most supportive in the community** - helping others, adding links, tutoring those struggling

The goal of this approach is to support and reward expertise in different parts of the Machine learning engineering toolbox.

Group Work Policy

Everyone has to submit all their work individually.

Late Submission Policy

Our goal is to prepare successful learners for a global level job. At work, deadlines are sometimes very strict - either you do it before the deadline or the company loses a substantial opportunity. Moreover, the late communication behavior (submission in 10 Academy can be considered as progress communication to team leads), blinds team leads and CEOs and is very determinantal in hindering the success of the company.

We have set our late submission as follows

- Submissions are accepted only within the 12 hrs window - 17:00 UTC - 7:00 UTC of the submission deadline
- Frequently late submissions (exceeding 6 total late submissions) will disqualify a person from the list of trainees 10 Academy recommends to partner employers.
- Badges will be rewarded for the cumulative on-time appearances (gmeet calls, on-time assignment submissions, and other places where being on-time is important)

From week 8 onwards, your two lowest weeks' scores will not be considered.

Instruction:

Objectives: Build Contract Optimised Q&A System

Build, Evaluate and Improve a simple RAG system for Contract Q&A (chatting with a contract and asking questions about the contract).

There are various frameworks for LLM apps, as well as RAG focused open source projects that simplify the creation of a RAG pipeline, such as Langchain, LlamaIndex and Azure Rag. Many companies build their own systems from scratch, but in this challenge, in order to focus on the essence (learning the RAG basics, evaluating the RAG pipeline and improving the quality of contract Q&A), we'll use Langchain - a leading LLM application framework.

Task 1: Research ways to improve RAG systems in general.

Task 1.1: Literature Review and Trend Analysis

- Review academic papers
    - Retrieval-Augmented Generation for Large Language Models: A Survey
    - Exploring the Latest Advancements in RAG Technology
    - G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment
- Document the challenges and possible solutions associated with building reliable RAG systems.
    - A Guide on 12 Tuning Strategies for Production-Ready RAG Applications | by Leonie Monigatti | Towards Data Science

Task 1.2: Build intuition on the various RAG performance metrics

- Understand the key metrics uses to measure RAG performance
    - How to Improve RAG Model Performance with Synthetic Data (gretel.ai)
- Experiment the factors that lead to poor and excellent performance
    - Use rag-datasets/mini-bioasq · Datasets at Hugging Face or rag-datasets/mini_wikipedia · Datasets at Hugging Face dataset to experiment with varying RAG evaluation frameworks provided in Optimizing RAG Applications: A Guide to Methodologies, Metrics, and Evaluation Tools for Enhanced Reliability | by Zilliz | Jan, 2024 | Medium
- Explore the impact of changing embedding models for retrievals and llm models for generating
    - Choosing the Right Embedding Model: A Guide for LLM Applications | by Ryan Nguyen | Medium

Task 1.3: Efficiency and Scalability

- Look for ways to optimize the speed and efficiency of both the retrieval and generation processes without compromising quality.

- Research architectural improvements that can help the system scale more effectively with larger datasets and increased user loads.

Task 1.4: Personalization and Contextualization in Generation

- Investigate how to incorporate user context and preferences to provide more personalized and relevant responses.
- Enhance the system's ability to understand and utilize context more effectively in both retrieval and generation.

Task 1.5: Bias Reduction

- Bias Detection and Mitigation: Develop methods to detect and reduce biases in responses generated by the RAG system.

Task 2: Build simple Q&A pipeline with RAG using Langchain

Task 2.1: Planning and Design

- Choose a Large Language Model: Research and select a suitable large language model based on factors like performance, scalability, and cost.
- Design Component Interaction:
    - Retriever and Generator: Plan how these two will share information and how the generator will use retrieved data.
    - Backend and Frontend: Outline the data flow between user interfaces and server-side processes. Consider user experience, data formats, and security aspects.

Task 2.2: Development of Each Component

- **Retriever Development:**
    - Choose between dense, sparse, or hybrid retrievers based on your data and requirements.
    - Customise training with your specific dataset to enhance retrieval relevance.
    - Perform thorough testing, and consider using A/B testing to refine the model.
- **Generator Development:**
    - Consider models that can generate coherent and contextually relevant answers.
    - Adapt the model to your domain-specific data for better performance.
    - Ensure the generator maintains accuracy and relevance in its responses.

Task 2.3: Integration and Testing

- **API Design and Endpoint Definition:**
    - Develop APIs that are robust and can handle varying loads.
    - Clearly define endpoints with proper documentation for ease of frontend integration.

- **Retriever and Generator Integration:**
    - Focus on the efficiency of data transfer between these components.
    - Implement error handling and fallback mechanisms.
- **Database and Storage**:
    - Choose databases that support quick read/write operations.
    - Implement encryption and other security measures to protect data.
- **Performance Optimization:**
    - Analyse bottlenecks and implement solutions like load balancing, query optimization, and efficient caching.

Task 3: Build a RAG evaluation pipeline with RAGAS

Task 3.1: Set Up Evaluation Metrics and Benchmarks

- Choose metrics that accurately measure the performance of your RAG system, such as accuracy, relevance, response time, and consistency.
- Define benchmarks or standards for comparison. This might include comparing with other models or predefined performance targets.

Task 2.2: Implement Evaluation Tools

- Determine the tools and frameworks needed for evaluation. Consider using existing libraries that can facilitate RAG evaluation.
- Ensure the evaluation tools can seamlessly interface with your RAG system for continuous assessment.

Task 3.3: Data for Evaluation

- Evaluation set which contains two Contracts (a short one and a long one) with a list of 10 questions and correct answers for each.

Task 4.4: Execution of Evaluation Pipeline

- Develop scripts or procedures for automated testing of the RAG system. Automating the process ensures consistency and efficiency in evaluation.
- Implement a manual review process for a qualitative assessment of the system's outputs.

Task 3.5: Analysis and Reporting

- Analyse the results from the evaluations to identify patterns, strengths, and weaknesses in the system.
- Create detailed reports outlining the performance of the RAG system. Include both quantitative metrics and qualitative insights.

Task 4: Idea to optimize Contract Q&A

Come up with a list of the top 5 tasks you suggest starting with in order to optimise for Contract Q&A. Be able to deeply explain these. Some inspiration in this regard are

- Deep Understanding of Legal Texts
    - Enhance the system's ability to process and understand legal language, which often includes complex and specific terminology.
    - Improve the system's capability to interpret the context in legal documents, which is crucial for accurate responses.
- Precision and Reliability
    - Implement techniques to increase the accuracy of responses, as precision is critical in legal contexts.
    - Establish methods to validate and cross-check the generated answers for reliability.
- Data Collection and Curation
    - Gather and curate extensive legal datasets, focusing on contracts and legal documents, to train the system.
    - Keep the dataset updated with the latest legal documents and changes in law to ensure relevancy.
- Custom Model Development
    - Develop or adapt retrieval models specifically for legal texts, optimising them for the unique structure and content of contracts.
    - Fine-tune generative models to produce responses that are not only accurate but also adhere to legal writing standards.

Task 5: Implement at least two enhancements

- Implement at least two enhancements,
- reevaluate each enhancement
- provide insights as to incremental improvement achieved with each enhancement (select the enhancements which you believe can make the highest impact given the time you have for the challenge).

Task 6: Interpretation & Reporting

- Present the challenge and your outcomes with a short deck and be ready to answer questions. We mainly look for a deep understanding of the task and its implementation phases.
- Report should have to include the system's performance metrics to quantify the incremental improvements.

Tutorials Schedule

Overview

- Monday:
- Tuesday:
- Wednesday:
- Thursday:

In the following, the colour **purple** indicates morning sessions, and **blue** indicates afternoon sessions.

Monday

Going through the main concepts behind evaluating and improving a simple RAG system for Contract Q&A.

- Challenge going through QA (Arnon from LizzyAI)
- How hard is the current week challenge? (Daniel from LizzyAI)

Wednesday

- Q&A with tutors

Deliverables

NOTE: Document should be a PDF stored in google drive or published blog link. **DO NOT SUBMIT A LINK as PDF!** If you want to submit a pdf document, it should be the content of your report not a link.

Interim Submission - Wednesday 8pm UTC

- Link to your code in GitHub
- Share a report about your project understanding and workflow. Maximum of 3 pages - PDF format please. Prepare this in a format that a 3rd-year student at a university can understand the basic concepts and reproduce your work.

Feedback

You may not receive detailed comments on your interim submission, but will receive a grade.

Final Submission - Saturday 8pm UTC

- Link to your code in GitHub
- A blog post entry (which you can submit for example to Medium publishing) in the form of a PDF report.

Feedback

You will receive comments/feedback in addition to a grade.

References

About machine learning

- What is machine learning and how does it work? In-depth guide
- What is natural language processing (NLP)?

About RAG

- The Future Unveiled: Advancements in RAG Technology with Canopy
- Retrieval Augmented Generation
- Use of RAG in Generative AI
- Latest Advancements in Retrieval Augmented Generation Technology Since 2023

Research papers

- Retrieval-Augmented Generation for Large Language Models: A Survey
- Exploring the Latest Advancements in RAG Technology

About RAG Evaluation

- Efficiently evaluating LLMs for legal tasks
- Evaluating Large Language Models (LLMs): A Standard Set of Metrics for Accurate Assessment
- Prompt benchmark
- Evaluations large language model
- CoQA Understand a text passage and answer a series of interconnected questions that appear in a conversation.
- Evaluating RAG Applications with RAGAs
- An Overview on RAG Evaluation
- Retrieval-Augmented Generation for Large Language Models: A Survey
- RAG System: Metrics and Evaluation Analysis with LlamaIndex
- RAG Evaluation Using LangChain and Ragas
- Evaluating the Performance of Retrieval-Augmented LLM Systems

About API design

- Best Practices in API Design