10 Academy Cohort A

Weekly Challenge: Week 6

Precision RAG: Prompt Tuning For Building Enterprise Grade RAG Systems

Business objective

PromptlyTech is an innovative e-business specializing in providing AI-driven solutions for optimizing the use of Language Models (LLMs) in various industries. The company aims to revolutionize how businesses interact with LLMs, making the technology more accessible, efficient, and effective. By addressing the challenges of prompt engineering, the company plays a pivotal role in enhancing decision-making, operational efficiency, and customer experience across various industries. PromptlyTech's solutions are designed to cater to the evolving needs of a digitally-driven business landscape, where speed and accuracy are key to staying competitive.

The company focuses on key services: Automatic Prompt Generation, Automatic Evaluation Data Generation, and Prompt Testing and Ranking.

1. Automatic Prompt Generation Service:

- This service streamlines the process of creating effective prompts, enabling businesses to efficiently utilize LLMs for generating high-quality, relevant content. It significantly reduces the time and expertise required in crafting prompts manually.

2. Automatic Evaluation Data Generation Service:

- PromptlyTech's service automates the generation of diverse test cases, ensuring comprehensive coverage and identifying potential issues. This enhances the reliability and performance of LLM applications, saving significant time in the QA(Quality Assurance) process.

3. Prompt Testing and Ranking Service:

- PromptlyTech's service evaluates and ranks different prompts based on effectiveness, helping Users to get the desired outcome from LLM. It ensures that chatbots and virtual assistants provide accurate, contextually relevant responses, thereby improving user engagement and satisfaction.

Background Context

In the evolving field of artificial intelligence, Language Models (LLMs) like GPT-3.5 and GPT-4 have become crucial for various applications. Their effectiveness, however, heavily depends on the quality of the prompts they receive, leading to the emergence of "prompt engineering" as a key skill.

Prompt engineering is the craft of designing queries or statements to guide LLMs to produce desired outcomes. The challenge lies in the sensitivity of these models to prompt nuances, where slight variations can yield vastly different results. This poses a significant hurdle for users, especially in business contexts where accuracy and relevance are paramount.

The need for simplified, efficient prompt engineering is clear. Automating and optimizing this process can save time, enhance LLM productivity, and make advanced AI capabilities more accessible to a broader range of users. The tasks of Automatic Prompt Generation, Evaluation Data Generation, and Prompt Testing and Ranking are aimed at addressing these challenges, streamlining the prompt engineering process for more effective use of LLMs.

Learning Outcomes

Skills Development

- Prompt Engineering Proficiency: Gain expertise in crafting effective prompts that guide LLMs to desired outputs, understanding nuances and variations in language that impact model responses.
- Critical Analysis: Develop the ability to critically analyze and evaluate the effectiveness of different prompts based on their performance in varied scenarios.
- Technical Aptitude with LLMs: Enhance technical skills in using advanced language models like GPT-4 and GPT-3.5-Turbo, understanding their functionalities and capabilities.
- Problem-Solving and Creativity: Cultivate creative problem-solving skills by generating innovative prompts and test cases, addressing complex and varied objectives.

- Data Interpretation: Learn to interpret and analyze data from test cases and prompt evaluations, deriving meaningful insights from performance metrics.

Knowledge Acquisition

- Understanding of Language Models: Acquire a deeper understanding of how LLMs function, including their strengths, limitations, and the principles behind their responses.
- Insights into Automated Evaluation Data Generation: Gain knowledge about the methodology and importance of creating test cases for evaluating prompt effectiveness.
- ELO Rating System and its Applications: Learn about the ELO rating system used for ranking prompts, understanding its mechanics and relevance in performance evaluation.
- Prompt Optimization Strategies: Understand various strategies for refining and optimizing prompts to achieve better alignment with specific goals and desired outcomes.
- Industry Best Practices: Familiarize with the best practices in prompt engineering within different industries, learning about real-world applications and challenges.

Team

Tutors:

- Yabebal
- Emitinan
- Rehmet

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.

Instruction: Automatic Prompt Engineering

Fundamental Tasks
The core tasks for this week's challenge in Automatic Prompt Engineering are outlined below:

1. Understand Prompt Engineering Tools and Concepts: Gain a thorough understanding of the tools and theoretical concepts involved in prompt engineering for Language Models (LLMs).

2. Familiarize with Language Models: Learn about the capabilities and functionalities of advanced LLMs like GPT-4 and GPT-3.5-Turbo.

3. Develop a Plan for Prompt Generation and Testing: Create a comprehensive plan that outlines the approach for automated prompt generation, test case creation, and prompt evaluation.

4. Set Up a Development Environment: Prepare a suitable development environment that supports the integration and testing of LLMs in the prompt engineering process.

5. Design User Interface for Prompt System: Plan and initiate the development of a user-friendly interface for prompt input, refinement, and performance analysis.

6. Plan Integration of LLMs: Strategize the integration of LLMs into the prompt system for automated generation and testing.

7. Build and Refine Prompt Generation System: Develop the automated prompt generation system, ensuring it aligns with user inputs and objectives.

8. Develop Automatic Evaluation Data Generation System: Create a system for generating test cases that evaluate the effectiveness of prompts in various scenarios.

9. Implement Prompt Testing and Evaluation Mechanism: Set up testing procedures using Monte Carlo matchmaking and ELO rating systems to evaluate and rank prompts.

10. Refine and Optimize System Based on Feedback: Continuously refine the prompt generation and evaluation system based on user feedback and performance data.

Task 1: Review the Evolution of Automatic Prompt Engineering

Focus on understanding the key developments in the field of automatic prompt engineering for Language Models (LLMs).

Study Key Concepts and Tools:

- Understand the key components of an enterprise grade RAG systems
  - Retrieval-augmented generation (RAG): What it is and why it's a hot topic for enterprise AI
  - Advanced RAG for LLMs/SLMs
  - RAG for Text Generation Processes in Businesses (check part 1, 3, & 4 as well)
  - Langchain Reterivers
- Understand the need for advanced prompt engineering in building enterprise grade RAG systems
  - Full Fine-Tuning, PEFT, Prompt Engineering, and RAG: Which One Is Right for You?
  - Advanced Prompt Engineering - Practical Examples
  - Prompt Engineering 201: Advanced methods and toolkits
  - Do you agree with this article? RAG is Just Fancier Prompt Engineering
- Understand the need for evaluating RAG components
  - An Overview on RAG Evaluation
  - Evaluating RAG: Using LLMs to Automate Benchmarking of Retrieval Augmented Generation Systems
  - Evaluating RAG Applications with RAGAs
  - RAG Evaluation Using LangChain and Ragas
  - RAG System: Metrics and Evaluation Analysis with LlamaIndex
  - Evaluating RAG Part I: How to Evaluate Document Retrieval
  - Evaluating RAG/LLMs in highly technical settings using synthetic QA generation
  - Evaluating Multi-Modal RAG
- Understand the tools and techniques to automatically generate RAG evaluation data
  - The Tech Buffet #16: Quickly Evaluate your RAG Without Manually Labeling Test Data
  - Generating a Synthetic Dataset for RAG
  -
- Learn key packages to planning, building, testing, monitoring, and deploying enterprise grade RAG system
  - Iterate on LLMs faster: Measure LLM quality and catch regressions
  - Building RAG-based LLM Applications for Production
  - ARES: An Automated Evaluation Framework for Retrieval-Augmented Generation Systems
- Understand the end-to-end technology stack of RAG systems
  - End-to-End LLMOps Platform

- An Enterprise-Grade Reference Architecture for the Production Deployment of LLMs Using the RAG Pattern on Azure OpenAI

Task 2: Design and Develop the Prompt Generation System

- Users can input a description of their objective or task and specify a few scenarios along with their expected outputs.
- Write or adopt sophisticated algorithms, you generate multiple prompt options based on the provided information.
- This automated prompt generation process saves time and provides a diverse range of alternatives to consider. But add an evaluation metrics that check whether the generated prompt candidate aligns with the input description.

Task 3: Implement Evaluation Data Generation and Evaluation

To further enhance the prompt generation process, incorporate automatic Evaluation Data Generation.

- By analysing the description provided by the user, create a set of test cases that serve as evaluation benchmarks for the prompt candidates.
- These test cases simulate various scenarios, enabling users to observe how each prompt performs in different contexts.
- The generated test cases serve as a starting point, sparking creativity and inspiring additional test cases for comprehensive evaluation.

Task 4 : Prompt Testing and Ranking

Goals

Comprehensive Evaluation: Provide a robust system that uses various methodologies for a thorough assessment of prompts.

Customizable and User-Centric: Allow users to choose or customize their preferred evaluation methods.

Dynamic and Adaptive: Ensure the system remains flexible and adaptive, capable of incorporating new ranking methodologies as they emerge.

Primary Methods

- **Monte Carlo Matchmaking:** This method is used to select and match different prompt candidates against each other. The Monte Carlo method, known for its applications in problem-solving and decision-making processes, helps in optimizing the information gained from each prompt battle. By simulating various matchups, it allows the system to test the effectiveness of each prompt in different scenarios.
- **ELO Rating System:** This system, which is commonly used in chess and other competitive games, rates the prompts based on their performance in the battles. Each prompt candidate is assigned a rating that reflects its success in previous matchups. The system takes into account not just the number of wins but also the

strength of the opponents each prompt has defeated. This rating helps in objectively ranking the prompts based on their effectiveness.

Additional Ranking and Matching Mechanisms

- **TrueSkill Rating System**: Ideal for scenarios involving multiple competitors, adjusting ratings based on not just wins and losses but also the uncertainty in performance.
- **Glicko Rating System:** Similar to ELO but with added flexibility, accounting for the volatility in a player's (or prompt's) performance and the reliability of their rating.
- **Bayesian Rating Systems**: Applies Bayesian inference for a probabilistic approach to rating, considering uncertainties and contextual variations in prompt performance.
- **Pairwise Comparison Methods**: Involves direct comparisons between pairs of prompts, potentially integrating user preferences or expert evaluations into the ranking process.
- **Categorical Ranking**: Instead of a numerical rating, prompts are categorized based on performance criteria like creativity, relevance, etc., for more qualitative assessments.
- **Adaptive Ranking Algorithms**: Algorithms that learn and adjust over time, considering historical performance data and evolving user preferences or requirements.
- **Semantic Similarity Matching**: Using NLP techniques to match prompts based on semantic content, ideal for understanding nuanced differences in prompt effectiveness.

You should adopt an innovative approach to prompt evaluation by utilizing **Monte Carlo matchmaking** and **ELO rating systems,** or any alternative method to match and rank.

Task 5: User Interface Development

Develop a user-friendly interface for interacting with the prompt engineering system.

- UI Design: Plan and design a user interface that allows users to easily input data, receive prompts, and view evaluation results.
- UI Implementation: Develop and integrate the user interface with the backend prompt engineering system.

Task 6: System Integration and Testing

- Integrate all components of the system and conduct comprehensive testing.
- Integrate the prompt generation, Evaluation Data Generation, evaluation, and user interface components.
- Test the entire system for functionality, usability, and performance. Refine based on feedback and test results.

Tutorials Schedule

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

Monday: Understanding Prompt engineering

Here the trainees will understand the week's challenge.

- Introduction to Week Challenge (Yabebal)
- Introduction and challenge to prompt engineering (Fikerte)

Key Performance Indicators:

- Understanding week's challenge
- Understanding the prompt engineering
- Ability to reuse previous knowledge

Tuesday

- RAG components (Rehmet)
- Techniques to improving R (Retrievers) in RAG (Emitnan)

Key Performance Indicators:

- Understanding Prompt ranking
- Understanding prompt matching
- Ability to reuse previous knowledge

Wednesday

- RAG Evaluation Data Generation (Abel)
- Understanding of prompt matching and ranking (Mahlet)

Thursday

- RAG evaluation metrics (Emitnan)
- RAGObs - DevObs of RAG development and production deployment

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 pdf document, it should be the content of your report not a link.

Interim Submission - Wednesday 8pm UTC

- Link to your code in GitHub
    - Repository where you will be using to complete the tasks in this week's challenge. A minimum requirement is that you have a well structured repository and some coding progress is made.

- A review report of your reading and understanding of Task 1 and any progress you made in other tasks.

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
    - Complete work for Automatic prompt generation
    - Complete work for Automatic evaluation
    - Complete work for Evaluation Data Generation

- A blog post entry (which you can submit for example to Medium publishing) or a pdf report. .

Feedback

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

References

- Meistrari didn't see a good solution for prompt engineering, so it's building one
- AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts
- Large Language Models Are Human-Level Prompt Engineers
- Prompt Engineering
- How to Create a Monte Carlo Simulation using Python
- Monte Carlo Method Explained
- What is Monte Carlo Simulation? How does it work?
- Elo Rating Algorithm
- Elo algorithm implementation in Python
- TrueSkillTM: A Bayesian skill rating system