Escuela Politécnica Nacional
Facultad de Ingeniería en Sistemas
Carrera de Ingeniería en sistemas informáticos y de computación

Proyecto Final: Implementación de un Sistema RAG (Retrieval-Augmented Generation)

Integrantes: Carlos Córdova
Hernán Sánchez
Galo Tarapués

Informe de Avances del Proyecto: Implementación de un Sistema RAG

1. Introducción

El objetivo de este informe es evaluar el estado actual del **Sistema RAG para la consulta de planes de candidatos**, determinando si el progreso realizado hasta el momento cumple con los objetivos establecidos. Se analizarán los avances en cada fase del desarrollo, la efectividad de la planificación bajo la metodología **Scrum** y las mejoras necesarias para optimizar el rendimiento del sistema.

2. Estado Actual del Proyecto

2.1. Avance General

El desarrollo del **Sistema RAG para la consulta de planes de candidatos** ha seguido la planificación establecida bajo la metodología **Scrum**, alcanzando un estado funcional en sus principales componentes. Tanto el backend como el frontend responden correctamente, permitiendo la interacción con el usuario. Sin embargo, se han identificado áreas de mejora en la generación de respuestas, las cuales requieren optimización para garantizar mayor precisión.

En comparación con la planificación inicial, el desarrollo ha avanzado conforme a lo previsto en los sprints, logrando la integración de los módulos principales y ejecutando pruebas en cada fase. No obstante, se han redistribuido tareas en función de las necesidades detectadas durante la implementación.

2.2. Avance por Módulo

2.2.1. Obtención y Preprocesamiento del Corpus

- Se han recopilado documentos y transcripciones relevantes para la generación del corpus.
- Se implementó el preprocesamiento de datos, incluyendo limpieza, tokenización y eliminación de stopwords.

- Se generaron los embeddings utilizando **SentenceTransformers**, lo que permite representar semánticamente las descripciones.
- Estado actual: **Completado y funcional**.

2.2.2. Módulo de Recuperación

- Se implementó el sistema de recuperación basado en **FAISS**, permitiendo búsquedas eficientes mediante similitud de embeddings.
- Se realizaron pruebas para evaluar la calidad de los resultados obtenidos.
- Estado actual: **Funcional, pero requiere ajustes en la optimización de relevancia de los documentos recuperados**.

2.2.3. Módulo de Generación

- Se seleccionó un modelo de lenguaje basado en **transformers** para la generación de respuestas.
- Se realizaron pruebas iniciales, evidenciando que las respuestas aún pueden presentar imprecisiones.
- Estado actual: **Funcional, pero con oportunidades de mejora en la coherencia y precisión de las respuestas generadas**.

2.2.4. Evaluación del Sistema

- Se han aplicado métricas como precisión, recall y F1-score para evaluar el rendimiento del sistema de recuperación.
- Se han identificado áreas de mejora en la generación de respuestas, lo que sugiere la necesidad de ajustes en la formulación de los prompts o en la selección del modelo.
- Estado actual: **En progreso, con necesidad de ajustes en la evaluación de calidad de respuestas**.

2.2.5. Implementación Técnica e Interfaz

- El backend en **Python** responde correctamente a las consultas y gestiona la recuperación y generación de respuestas.
- La interfaz en **HTML, CSS y JavaScript** permite la interacción con el usuario de manera fluida.
- Estado actual: **Funcional, sin problemas críticos, pero con mejoras pendientes en la presentación de respuestas**.

3. Planificación y Estimación de Finalización

Comparación con la planificación inicial (cumplimiento de sprints): El progreso realizado ha sido satisfactorio, ya que los sprints establecidos han sido cumplidos dentro del tiempo previsto. Los módulos principales se han desarrollado con éxito, pero es importante destacar que algunos aspectos requieren optimización. A pesar de esto, no se han presentado retrasos significativos y el sistema sigue en línea con la planificación inicial.

Estimación del tiempo restante para completar cada módulo: Si no surgen problemas adicionales, se estima que el tiempo restante para completar cada módulo es adecuado. La optimización de la generación de respuestas y la mejora de la relevancia de los documentos recuperados son tareas que se pueden realizar dentro de los plazos establecidos para el Sprint 2.

4. Conclusión y Recomendaciones

Evaluación del cumplimiento de objetivos: El proyecto ha logrado establecer una estructura sólida que permite cumplir con los objetivos establecidos. Los avances realizados hasta ahora reflejan un progreso significativo, especialmente en el desarrollo de los módulos clave (obtención del corpus, recuperación de documentos y generación de respuestas). A pesar de algunos puntos que aún requieren ajustes, como la mejora en la precisión de las respuestas, el proyecto va por buen camino.

Recomendaciones para optimizar el desarrollo restante:

1. **Mejoras en la generación de respuestas**: Sería recomendable afinar el modelo de generación de respuestas para mejorar su coherencia y relevancia. Esto podría incluir la personalización de los prompts o la optimización del modelo de lenguaje utilizado, lo que aumentaría la calidad de las respuestas generadas.
2. **Optimización del sistema**: Aunque los módulos funcionan correctamente, las consultas a FAISS y la generación de embeddings pueden beneficiarse de una optimización adicional. Esto permitiría una mayor eficiencia en la recuperación de documentos y en la generación de respuestas, lo cual es crucial para el rendimiento del sistema a medida que crece el corpus.
3. **Mejor presentación de respuestas**: Mejorar la presentación de las respuestas en la interfaz del usuario puede hacer que el sistema sea más fácil de usar y más accesible. Considerar la inclusión de elementos visuales o mejor formato de las respuestas puede contribuir a una mejor experiencia de usuario.

Escuela Politécnica Nacional
Facultad de Ingeniería en Sistemas
Carrera de Ingeniería en sistemas informáticos y de computación

Sprints

Sprint 1: Implementación Base

Duración: 27 de enero - 31 de enero (5 días)
Objetivo: Desarrollar la funcionalidad principal del sistema RAG, asegurando que el flujo de consulta y recuperación de documentos funcione correctamente.

Tareas y Responsabilidades:

Carlos Córdova (Backend & Embeddings):

- Implementar el preprocesamiento de datos, asegurando que los textos sean limpiados y tokenizados correctamente.
- Generar los embeddings para los documentos procesados y almacenarlos para su posterior uso.
- Configurar la base de datos y el índice FAISS para permitir búsquedas eficientes.

Hernán Sánchez (Backend & Generación de Respuestas):

- Implementar el módulo de recuperación de documentos basado en embeddings.
- Desarrollar la lógica de generación de respuestas, asegurando que el modelo pueda proporcionar información relevante a partir de los documentos recuperados.
- Integrar la API del backend con la generación de respuestas y garantizar que el servidor procese correctamente las solicitudes.

Galo Tarapues (Frontend & Integración General):

- Diseñar la interfaz de usuario para la consulta de información, asegurando una experiencia fluida y accesible.
- Integrar el frontend con el backend para mostrar los resultados obtenidos.
- Realizar pruebas iniciales para verificar el correcto funcionamiento del flujo de consulta.

Sprint 2: Optimización y Refinamiento

Duración: 1 de febrero - 7 de febrero (6 días)
Objetivo: Mejorar la precisión de las respuestas, optimizar el rendimiento del sistema y corregir errores detectados en la primera fase.

Tareas y Responsabilidades:

Carlos Córdova (Optimización del Backend & Embeddings):

- Mejorar la calidad de los embeddings y ajustar el modelo de recuperación para priorizar documentos más relevantes.
- Optimizar las consultas a FAISS para hacerlas más eficientes y reducir tiempos de respuesta.
- Implementar un manejo de errores robusto en la API para situaciones en las que no se encuentren documentos relevantes.

Hernán Sánchez (Mejoras en Generación de Respuestas):

- Refinar el modelo de generación de respuestas, asegurando que la información proporcionada sea más precisa y coherente.
- Evaluar el desempeño del sistema con métricas de precisión, relevancia y coherencia.
- Ajustar la arquitectura del backend según las pruebas realizadas y solucionar errores detectados.

Galo Tarapues (Optimización del Frontend & Evaluación de Usabilidad):

- Mejorar la presentación de resultados en la interfaz, haciendo que las respuestas sean más legibles y estructuradas.
- Realizar pruebas exploratorias para evaluar la experiencia del usuario y detectar posibles puntos de mejora.
- Reportar inconsistencias en la precisión de las respuestas y colaborar con el backend en la optimización del sistema.

Reuniones Scrum y Definición de “Hecho”

- **Daily Standups:** 15 minutos diarios para revisar avances y bloqueos.
- **Revisión y Retrospectiva:** Se realizan al final de cada sprint.
- **Definición de "Hecho":** Una tarea es considerada completa solo si está completamente implementada y funcional.