ESCUELA POLITÉCNICA NACIONAL
RECUPERACIÓN DE INFORMACIÓN
2024 – B

Memoria Técnica - Proyecto II Bimestre

Integrantes

Rossy Armendariz

Alejandro Chávez

Wilmer Rivas

Docente

Ing. Iván Carrera

Fecha de entrega

13-02-2025

1. Introducción

En este documento vamos a detallar la implementación del proyecto utilizando Scrum como metodología ágil, con el objetivo de estructurar el desarrollo de un sistema de recuperación de información basado en FAISS, BERT y modelos de lenguaje (LLM). Se documentan la planificación, los roles del equipo, la ejecución de los sprints y la evaluación final del proceso ágil.

2. Planificación Inicial del Proyecto

En la planificación inicial, se definieron los siguientes aspectos clave:

Objetivo: Desarrollar un sistema de consulta optimizado mediante FAISS y embeddings de BERT para recuperar información relevante de documentos de candidatos electorales.

Alcance: Implementar un pipeline de preprocesamiento, vectorización, indexación y generación de respuestas utilizando un modelo LLM.

Requerimientos iniciales: Uso de documentos en PDF y CSV, integración de FAISS para búsqueda eficiente y aplicación de técnicas de NLP para mejorar la generación de respuestas.

Tecnologías:

- Python para el procesamiento de datos y la implementación del modelo.
- FAISS para indexación y recuperación eficiente.
- BERT/Sentence-BERT para la generación de embeddings semánticos.
- Scrum como marco de trabajo para la organización del equipo.

3. Roles Asignados en el Equipo y Responsabilidades

Nombre	Rol	Responsabilidad
Alejandro Chávez	Scrum Master	Facilitar reuniones, asegurar el cumplimiento de Scrum, resolver impedimentos.
Rossy Armendariz	Developer, Tester	Implementación de código, pruebas funcionales, verificación de calidad, validación de respuestas del sistema, integración de componentes.
Wilmer Rivas	Developer, Tester	Implementación de código, integración de FAISS y embeddings, pruebas funcionales y optimización del pipeline.

4. Sprints Definidos y Retrospectivas

El desarrollo del proyecto se dividió en tres sprints con objetivos específicos.

Sprint 1: Preparación del Corpus y Preprocesamiento

Objetivo: Recolectar los documentos, preprocesar el texto y almacenar los datos en CSV.

Tareas clave:

- Conversión de PDF a texto.
- Eliminación de caracteres especiales, tokenización, lematización y stopwords.
- Generación de archivos CSV con oraciones procesadas.

Retrospectiva: Se optimizó el preprocesamiento y se redujo el ruido en los datos. Sin embargo, hubo retrasos debido a problemas con algunos documentos ilegibles.

Sprint 2: Vectorización e Indexación

Objetivo: Transformar el texto en embeddings y construir el índice FAISS.

Tareas clave:

- Generación de embeddings con BERT.
- Creación del índice FAISS para consultas eficientes.

Retrospectiva: FAISS mejoró significativamente los tiempos de búsqueda. Se decidió optimizar el índice utilizando un enfoque HNSW (Hierarchical Navigable Small World) para mayor rapidez.

Sprint 3: Integración con LLM y Generación de Respuestas

Objetivo: Utilizar un modelo de lenguaje para generar respuestas basadas en documentos recuperados.

- Tareas clave:
- Conversión de consultas en embeddings.
- Búsqueda en el índice FAISS y recuperación de oraciones más relevantes.
- Uso de GPT o BERT fine-tuned para generar respuestas coherentes.

Retrospectiva: Se lograron respuestas más precisas y contextualizadas. Sin embargo, se detectaron casos de sesgo en algunos resultados, lo que llevó a ajustar el preprocesamiento y la selección de documentos.

5. Documentación de Decisiones Tomadas Durante el Desarrollo

Durante la implementación, se tomaron las siguientes decisiones clave:

1. Uso de FAISS en lugar de búsquedas tradicionales:

Se optó por FAISS debido a su capacidad para manejar grandes volúmenes de datos y su rapidez en la recuperación de información.

2. Elección de embeddings con Sentence-BERT en vez de TF-IDF:

   Los embeddings permitieron capturar mejor la semántica de las consultas en comparación con una representación basada en palabras clave.

3. Corrección de sesgos en los datos:

   Se identificó que algunas respuestas favorecían ciertos enfoques debido a la composición del corpus, por lo que se realizó un ajuste manual de datos irrelevantes.

4. Uso de Ollama con llama3.2:latest para la generación de respuestas:

   Para la generación de respuestas a partir de los documentos recuperados, se eligió Ollama con el modelo llama3.2:latest debido a los siguientes factores:

   - Ejecución local optimizada, reduciendo dependencia de la nube y tiempos de inferencia.
   - Mejor comprensión de contexto, permitiendo generar respuestas más coherentes y relevantes a partir de los documentos seleccionados.
   - Fácil integración con FAISS, lo que permitió que el flujo de consulta-recuperación-generación sea más eficiente.

Justificación del modelo de generación:
Se compararon distintas opciones, y llama3.2:latest ofreció un balance entre velocidad y calidad de respuestas en pruebas iniciales. Su capacidad de generar texto basado en documentos extensos mejoró significativamente la coherencia de las respuestas generadas.

6. Evaluación Final del Uso de Scrum en el Proyecto

La implementación de Scrum permitió un desarrollo más organizado y colaborativo, con iteraciones claras y objetivos bien definidos. Los puntos positivos y áreas de mejora fueron los siguientes:

Puntos Positivos

- ☺ Organización efectiva del trabajo en sprints, permitiendo mejoras incrementales en cada iteración.
- ☺ El Scrum Master ayudó a mantener la coordinación y a solucionar bloqueos de manera ágil.

- ☺ Las retrospectivas permitieron ajustar estrategias en cada sprint.

Áreas de Mejora

- ❑ Se evidenció la necesidad de una mejor planificación en la asignación de tareas, ya que algunos procesos como la generación de embeddings tomaron más tiempo del esperado.
- ❑ Hubo dificultades en la validación de respuestas generadas, lo que resaltó la importancia de definir mejores métricas de evaluación automática en futuros proyectos.

En conclusión, scrum fue útil para organizar el trabajo y mejorar el sistema iterativamente. Sin embargo, en proyectos con IA generativa y NLP, es clave definir métricas de evaluación desde el inicio para optimizar los resultados.