udp

Tarea 3: Análisis Lingüístico Offline con Hadoop y Pig

Profesor: Nicolás Hidalgo
Ayudantes: Isidora González, César Muñoz Rivera, Natalia Ortega y Joaquín Villegas

Introducción y Contexto

En las entregas anteriores, el proyecto se ha centrado en la construcción de un sistema robusto para el procesamiento de solicitudes en tiempo real. Se implementó un pipeline de datos asíncrono (con Apache Kafka) capaz de gestionar fallos y un motor de análisis de streaming (con Apache Flink) para la evaluación de calidad y la mejora continua de las respuestas a nivel individual.

Tras estas etapas, nuestro sistema ha acumulado un valioso conjunto de datos históricos, compuesto tanto por las respuestas originales de Yahoo! como por las respuestas generadas por el LLM. El enfoque del proyecto se desplaza ahora del procesamiento online (evento a evento) al análisis offline (batch).

El objetivo de esta última entrega es explotar este repositorio de datos para realizar un análisis lingüístico a gran escala. Nos interesa responder preguntas como: ¿Qué vocabulario utiliza el LLM en comparación con las respuestas humanas? ¿Cuáles son las palabras más comunes en cada conjunto? ¿Existen patrones léxicos que diferencien una respuesta generada por IA de una humana?

Para realizar este análisis sobre un volumen de datos considerable, utilizaremos herramientas estándar del ecosistema Big Data: Apache Hadoop para el almacenamiento y procesamiento distribuido, y Apache Pig para crear scripts de alto nivel que implementen la lógica de MapReduce.

Entregable 3: Servicio de Análisis Batch de Vocabulario

El objetivo principal de esta tercera entrega es el diseño y desarrollo de un servicio de análisis batch que procese la totalidad de las respuestas almacenadas (humanas y del LLM) para extraer estadísticas y patrones lingüísticos.

Este servicio operará de forma independiente a los módulos de streaming desarrollados en la Tarea 2. Deberá ser capaz de conectarse a la base de datos (o a un dump de esta), extraer las respuestas, cargarlas en el ecosistema Hadoop y ejecutar trabajos de MapReduce (definidos mediante scripts de Pig) para realizar un conteo de palabras y un análisis de frecuencia.

El entregable se centrará exclusivamente en este nuevo módulo de análisis. No es necesario integrar ni entregar los servicios de Kafka, Flink o la API desarrollados en las fases anteriores.

Requerimientos y Organización del Sistema

Para la correcta evaluación de esta entrega, el sistema deberá cumplir con los siguientes objetivos específicos:

- **Ingesta de Datos:** El sistema debe ser capaz de extraer las respuestas de Yahoo! y las respuestas del LLM desde la base de datos persistente del proyecto (o un dump).
- **Ecosistema Hadoop:** Se debe implementar un entorno de Hadoop. El uso de HDFS para el almacenamiento de los datos de entrada y salida es fundamental.
- **Procesamiento con Apache Pig:** La lógica de análisis de texto debe ser implementada obligatoriamente usando Apache Pig. Los scripts de Pig deberán ejecutar un proceso MapReduce para el conteo de palabras.
- **Análisis de Frecuencia de Palabras:** El script de Pig debe realizar como **MÍNIMO** las siguientes tareas de procesamiento de texto:

  - Tokenización: Separar cada respuesta en palabras individuales.
  - Limpieza: Convertir el texto a minúsculas, eliminar signos de puntuación y filtrar stopwords comunes (ej. “el”, “la”, “que”, “de”, etc.).
  - Conteo: Calcular la frecuencia de cada palabra (WordCount).

- **Análisis Comparativo:** El análisis debe realizarse por separado para los dos conjuntos de datos:
  - Las respuestas de los usuarios de Yahoo!.
  - Las respuestas generadas por el LLM. El objetivo es poder comparar los resultados de ambos.

- **Distribución y Despliegue:** El servicio de análisis, incluyendo el entorno de Hadoop y Pig, deberá estar completamente containerizado utilizando Docker y orquestado mediante docker-compose.yml.

- **Foco del entregable:** Se reitera que la evaluación se centrará únicamente en este nuevo módulo de análisis batch. No se requiere la entrega de los componentes de la Tarea 2 (Kafka, Flink, consumidores, etc.).

- **Documentación y Buenas Prácticas**
  - Se deberá entregar una documentación técnica exhaustiva que abarque tanto la funcionalidad implementada como el código fuente[1]. Esta incluirá una justificación rigurosa de las decisiones de diseño, la elección de tecnologías y las metodologías de desarrollo aplicadas.

[1]Se requiere que el código esté alojado y documentado en un sistema de control de versiones, como GitHub o GitLab, siguiendo las buenas prácticas de la industria (ej. README.md detallado, comentarios en el código, etc.).

Análisis y Discusión

En esta sección, se debe realizar un análisis crítico y una discusión fundamentada sobre el diseño de la solución batch y los resultados lingüísticos obtenidos. El análisis debe estar respaldado por los datos generados y visualizaciones claras. Se deben abordar obligatoriamente los siguientes puntos:

1. Resultados del Análisis Lingüístico:

- Presentar los resultados del conteo de palabras. Se deben incluir tablas o gráficos (ej. nubes de palabras, gráficos de barras) que muestren el Top N (ej. Top 50) de las palabras más frecuentes para las respuestas de Yahoo! y para las respuestas del LLM. Cada gráfica debe ser justificada y explicada.

2. Discusión Comparativa:

- Comparar directamente ambos conjuntos de resultados. ¿Cuáles son las similitudes y diferencias más notables en el vocabulario?
- ¿Utiliza el LLM un vocabulario más formal, más simple o diferente al de los usuarios humanos?
- ¿Qué patrones o sorpresas encontraron al comparar las palabras más usadas?

3. Justificación de la Arquitectura (Pig y MapReduce):

- Justificar la elección de Pig para esta tarea. ¿Cómo facilitó Pig la implementación de la lógica de MapReduce en comparación con escribir un programa de MapReduce nativo en Java?
- Detallar los desafíos encontrados en el procesamiento de los datos (ej. limpieza de texto, manejo de caracteres especiales, definición de stopwords).

4. Discusión: Procesamiento Batch vs. Streaming:

- Basado en la experiencia de la Tarea 2 (Flink) y la Tarea 3 (Hadoop/Pig), discutir las diferencias fundamentales entre una arquitectura de streaming (online) y una de batch (offline).
- ¿Por qué este análisis de frecuencia de palabras era más adecuado para un modelo batch (Hadoop) que para un modelo de streaming (Flink)?

Requisitos de la Entrega

La evaluación de este hito se basará en los siguientes entregables:

- **Informe Técnico:** Un documento en formato LaTeX (entregado como PDF) que describa de manera concisa y rigurosa la arquitectura y el enfoque del sistema. El informe debe centrarse en el **análisis crítico** que justifique las decisiones tomadas y los resultados obtenidos. El contenido debe incluir:
  - Una descripción detallada de los componentes y las funcionalidades implementadas en cada módulo del sistema.
  - Una justificación fundamentada de las decisiones de diseño, así como de las tecnologías y metodologías empleadas.
  - Un análisis exhaustivo de los resultados, respaldado por datos empíricos. Se deben utilizar gráficos, tablas comparativas y otros recursos visuales para facilitar la interpretación y evaluación de las métricas definidas.
- **Video de Demostración:** Un video con una duración de 10 minutos donde se muestre el sistema en funcionamiento, explicando sus componentes y el flujo de datos.
- **Código Fuente:** El código fuente completo del proyecto, el cual deberá estar alojado en un repositorio público en GitHub o GitLab. El enlace al repositorio deberá incluirse tanto en la sección de comentarios de la plataforma CANVAS como en el informe técnico.
- **Archivos de Despliegue:** Un archivo `Dockerfile` y/o `docker-compose.yml` que permita la construcción y ejecución de todos los servicios implementados. El archivo `README.md` del repositorio deberá contener instrucciones claras y precisas para el despliegue y la ejecución del sistema.

Reglas y consideraciones de la entrega

- **Fecha de Entrega:** La fecha límite para la entrega de esta tarea es el día 21/10/2025 hasta las 23:59 hrs. Se recomienda gestionar el tiempo de forma efectiva. La entrega es vía Canvas del curso.
- **Integrantes:** La tarea debe ser realizada en grupos de hasta **2** alumnos/as.
- **Ética y Autoría:** Se debe respetar el reglamento de la universidad en cuanto a plagio y autoría. Cualquier evidencia de copia o falta de autoría conllevará sanciones según lo establecido.