Final Group Project

Updated August 11th, 2023

1 Introduction

One can argue that the most challenging task in a Big Data setting is getting the data that can then be used for data analysis and predictions. Towards this goal, in this project, you will be setting up a pipeline to ingest data from ~~twitter~~ **NewsAPI** using Apache Kafka, clean and process it, and load it into a Hive table for analysis. You can also optionally use a large language model (LLM) to gain further insights or transformations on the data.

2 Instructions

2.1 Step 0: Review theory and Setup Kafka Broker

Review videos on Apache Kafka and Apache Hive including the hands-on exercises. Setup a Kafka broker with Zookeeper as discussed in class.

2.2 Step 1: Get NewsAPI token and run sample python code

NewsAPI is a company that has archive of open source news articles from different sources. You can get a free API token to retrieve articles. With the free tier, there are limitation of 100 calls per day and you can only get articles for the past month but this is not an issue for this academic project. If you can't sign up for a token, use the sample token posted on the course shell.

Once you have the API token, run the sample python notebook in either Google colab or Jupytor notebook. See how the data is being read and what kind of data is returned from the NewsAPI. Experiment with some Python code of your own to play around with the data that is returned.

2.3 Step 1: Setup Kafka producer to ingest ~~tweets~~ articles

Setup a Kafka producer in Python that gets data from new articles for a specific set of keywords related to a topic (the choice of topic and keywords are up to

you). Send the data to a topic in a Kafka broker. The data should be formatted in a way that can be easily ingested by the other components of the pipeline.

A sample python producer is available to you for reference.

2.4 Step 2: Setup Kafka Consumer to read ~~tweets~~ articles

Setup a Kafka consumer that reads from the Kafka topic and saves the data to HDFS. The data that is saved needs to be cleaned and put into multiple columns. It is up to you how you want to clean the data, either in the consumer or producer for Kafka. You should ensure that the data is formatted in a way that can be easily loaded into Hive for analysis (see below).

2.5 Step 4: Load Data into a Hive Table

Data then must be loaded into a Hive table, and some queries run on it. You should write HiveQL queries to analyze the data and provide insights into the ~~tweets~~ articles that you have collected. The queries should be informative and insightful and hence you must think about the best to load the data into Hive. Do not just load the ~~tweets~~ articles as is into Hive, try to get some columns based on the ~~tweets~~ articles (this transformation must be done at the consumer or producer level) and write some insightful queries.

2.6 Step 5: Optional - Read ~~tweets~~ articles into an LLM

There are many large language models available the most famous of them being ChatGPT. Take the ~~tweets~~ articles that you have read and feed them to a LLM model which will then output something interesting. Examples include: Sentiment analysis, NER (name entity recognition), image generation, etc. Many LLM models exists from HuggingFace which we will briefly discuss in class.

2.7 Step 6: Optional - Create a Web Application

Create an web interface application using Gradio.app or streamlit.io that allows the user to click a button, get some ~~tweets~~ articles and then see the results of the LLM output (for example, if the ~~tweets~~ articles are positive or negative). An example web application will be demonstrated in class.

3 Deliverable

1. Programming files used for setting up the tools
2. A short report on your setup pointing out any technical difficulties and how you overcame them.
3. An *up to* 15 minute video showing a LIVE demo of the system working. Note that the live demo must show each component getting data live,

saving it to HDFS (or local file system) and you running a sample Hive Query on the data that is loaded. You should explain your code in the demo and point out any difficulties you had and how you overcame them.

4 Marking Scheme

The grading for this assignment will be distributed as follows:

- **Video presentation (30%)**: A video recording effectively demonstrating your system. If working in a team, all members must participate and explain some technical part of the pipeline. If a member is not present in the video, there is no marks given for the project.

  If there is no video presentation, there is no marks given for the project.

- **Correct ETL process setup (30%)**: You have correctly setup a Kafka broker, and have a consumer and producer running ingesting data.

- **Code and Explanations (10%)**: The code for producer and consumer should not just be what is given in examples provided. It should properly filter ~~tweet~~ article data and be explained. These days with ChatGPT, I see students just showing a bunch of code and saying "oh it runs". You need to clearly demonstrate that you actually understand the code through proper explanations.

- **Report (10%)**: A report that clearly highlights the process and has screenshots supporting that work was done.

- **Quality of work (15%)**: How well the system and ETL is designed and presented given the number of people in the group.

- **Discretionary (5%)**: Discretionary marks relate to the presentation, following steps, and generally things that are not covered in items above.

As mentioned above, failure to submit a video recording or submitting a video recording without any verbal explanation will result in a zero in the project (no part marks for submitting code etc). The reason for this is that the video is pivotal in demonstrating that you have done the work.