Lab 04: Streaming Data Processing with Spark

Result

Section	%	Note
1. Get Twitter tweets	100%	Using MongoDB to store database
2. Stream tweets to Apache Spark	100%	Using Apache Kafka for Streaming
3. Perform sentiment analysis on tweets	100%	
4. Visualize the analytic results	70%	Using Plotly and Plotly Dashboard

Introduction

Name Dang Huynh Cuu Quan.
Id is 20120354
He was assigned to do part 4 "Visualize the analytic results" and I have done by myself.

Name Nguyen Viet Khoa.
Id is 20120120
He was assigned to do part 3 "Perform sentiment analysis on tweets" and I have done by myself.

Name Nguyen Quang Tuyen.
Id is 20120120
He was assigned to do part 1 "Get Twitter tweets" and part 2 "Stream tweets to Apache Spark".

Name Nguyen Dinh Tri
Id is 20120218
He was assigned to do report.

Environment

My team do this lab on WSL2 but it can be done on any Linux environment

Part 1 Get Twitter tweets

Set up virtual environment using conda

Download miniconda

Create an `env` by running:

```
conda create --name lab4 python=3.10
```

Remember to activate `env` before running any python scripts

```
conda activate lab4
```

Import data to mongodb

First download dataset:

```
wget https://huggingface.co/datasets/deberain/ChatGPT-Tweets/resolve/main/train.csv
```

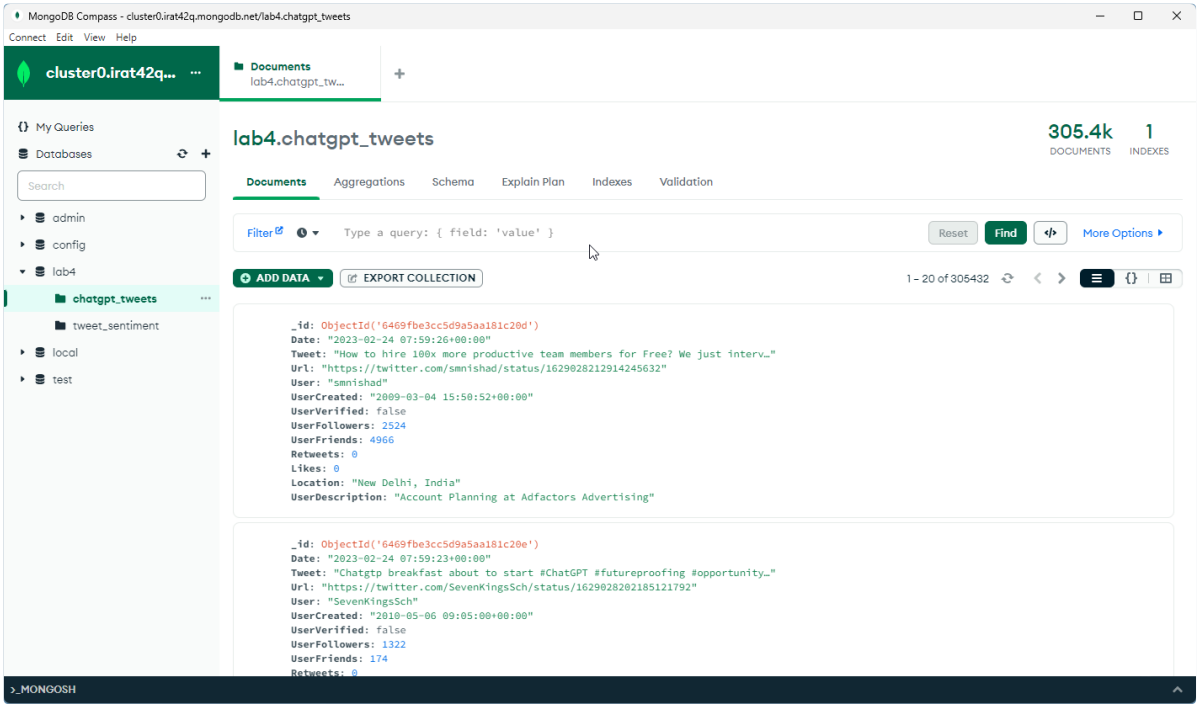
Download dependencies:

```
conda install -c conda-forge python-dateutil
conda install pymongo
python3 -m pip install pymongo[srv]
```

Then run the file `get_tweets.py` to parse CSV and import data to mongodb.

```
python3 get_tweets.py
```

Our team created an online mongodb cluster at `mongodb+srv://nvkhoa14:UITHKT@cluster0.irat42q.mongodb.net/` and collections can be view on MongoDB Compass



⌈ Part 2: Stream tweets to Apache Spark using Apache Kafka ⌋

Download pyspark, kafka

Download pyspark

```
conda install -c conda-forge pyspark
```

Download kafka

```
wget https://downloads.apache.org/kafka/3.4.0/kafka_2.13-3.4.0.tgz
tar -xzf kafka*
mv kafka*/ kafka
```

Start Kafka broker

Kafka with ZooKeeper

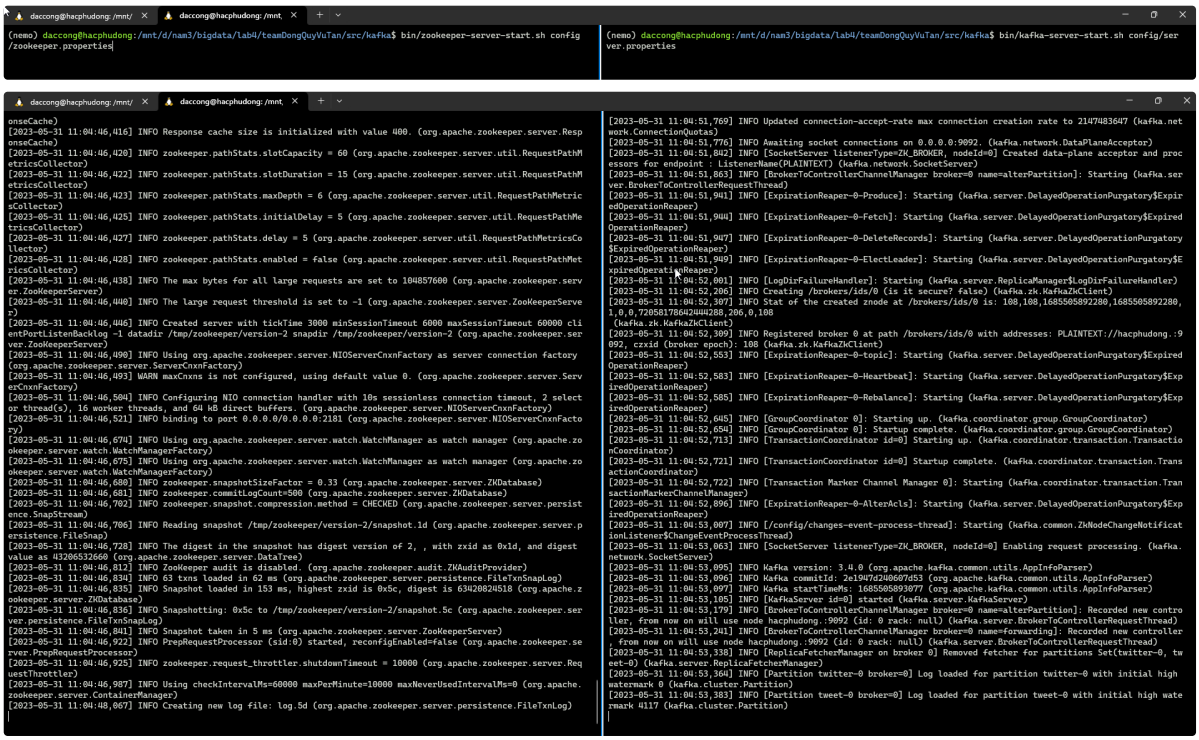
Open a terminal and go to kafka/ folder

Start the ZooKeeper service:

```
bin/zookeeper-server-start.sh config/zookeeper.properties
```

Open another terminal session start kafka broker service:

```
bin/kafka-server-start.sh config/server.properties
```



Feed data to kafka's topic

Run the file `test_producer.py` to feed data to kafka's topic.

```
python3 test_producer.py
```

It will first request data from mongodb, map it into a json form of `{message: tweet}` and send to topic named `tweet`

We can check if topic is written correctly by run a consumer:

```
bin/kafka-console-consumer.sh --topic tweet --from-beginning --bootstrap-server localhost:9092
```

Read data stream from pyspark

Now data can be read from pyspark by subscribe to topic `tweet`

```
spark = (
    SparkSession.builder
        .appName("TwitterSentimentAnalysis")
        .config("spark.mongodb.input.uri", CONNECTION_STRING)
        .config("spark.mongodb.output.uri", CONNECTION_STRING)
        .config("spark.jars.packages", "org.mongodb.spark:mongo-spark-connector_2.12:9.1.7")
        .config("spark.jars.packages", "org.apache.spark:spark-sql-kafka-0-10_2.12:3.4.0")
        .getOrCreate()
)

spark.sparkContext.setLogLevel("ERROR")

df = (
    spark
    .readStream
    .format("kafka")
    .option("kafka.bootstrap.servers", "localhost:9092")
    .option("subscribe", "tweet")
    .load()
)
```

⌈ Part 3: Perform sentiment analysis on tweets⌋

Analyze and output to console

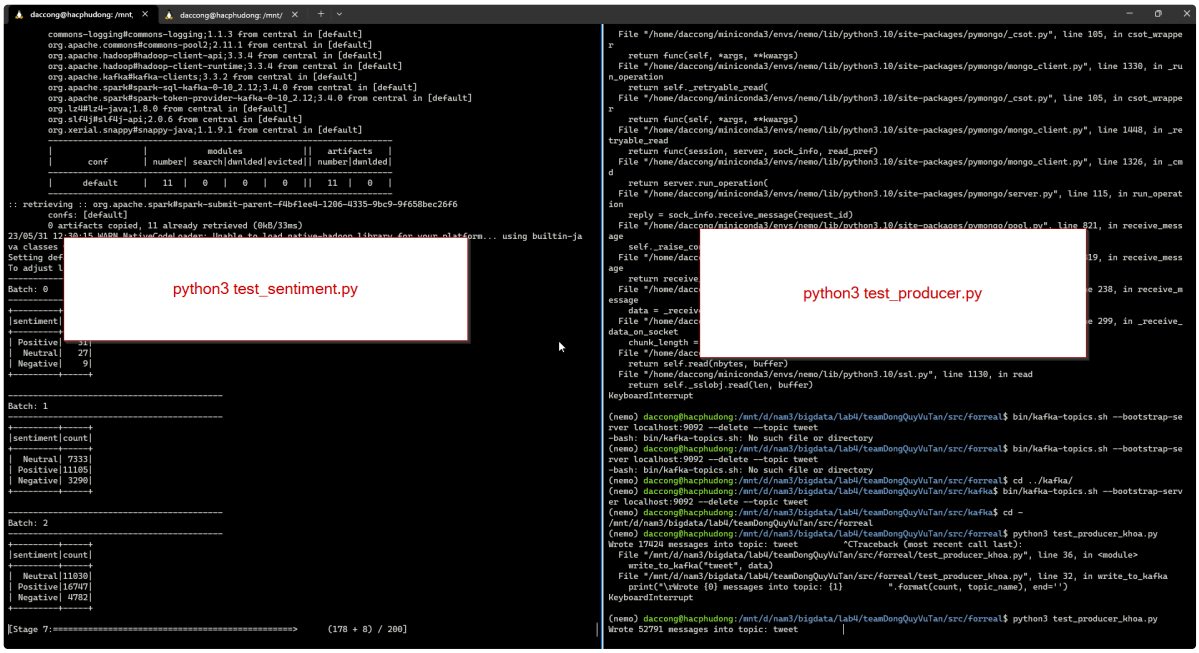
Download `textblob`

```
conda install -c conda-forge textblob
```

Run the file `test_sentiment.py` to start listening to newest topic's changes, analyzing tweets and output to *console*

```
python3 test_sentiment.py
```

We have to run `test_producer.py` after starting `test_sentiment.py` because it don't read topic's data from beginning



Save results to mongodb

We planned to save results to mongodb every batch so Plotly can read it and update the plot.

```
CONNECTION_STRING =
"mongodb+srv://nvkhoa14:UITHKT@cluster0.irat42q.mongodb.net/lab4.sink"

def write_row(batch_df , batch_id):

 df.write.format("mongo").mode("append").option("uri",CONNECTION_STRING).save()
    pass

query = (
    sentiment_tweets.groupby("sentiment").count().writeStream
    .foreachBatch(write_row)
    .start().awaitTermination()
)
```

Error

We got the error **"pyspark.errors.exceptions.captured.AnalysisException: Append output mode not supported when there are streaming aggregations on streaming DataFrames/DataSets without watermark;"** while saving results to mongodb and cannot fix it.

Part 4: Visualize the analytic results

In this section, our team use Plotly in order to create dynamic plot of sentiment trends over time. We have 2 approach for using it:

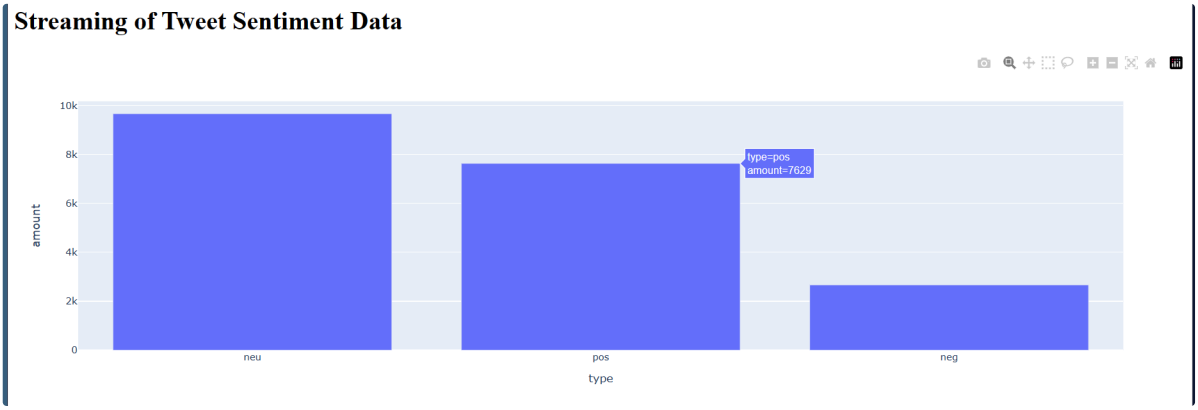
- dashboard.py the main file in ploting, directly get information in MongoDB over time while streaming process updates in mongo database.

- Beside dashboard.py, server.py act as a intermediate factor to interact between Spark app and Dashboard. Spark app will send the data to server through POST method and Dashboard get it through GET methods.

In this case, we implement the first approach. Run

```
python dashboard.py
```

to hosting dashboard server on web.



We can see that tweets tends to be NEU.

References

- For getting Twitter tweets and performing sentiment analysis on tweets: https://medium.com/@lorenagongang/sentiment-analysis-on-streaming-twitter-data-using-kafka-spark-structured-streaming-python-part-b27aecca697a?fbclid=IwAR1R615kSN4taU9O5d0YQGhPtCrvFNUpJPQFhMUZUKcp8eQ8osM_8KOzpRA
- Dash Plotly: https://dash.plotly.com/minimal-app
- Plotly: https://plotly.com/python/

Apache Kafka tutorials:

- For understanding Apache Kafka and setting up https://kafka.apache.org/documentation/streams/ https://developer.confluent.io/what-is-apache-kafka/
- For understanding some main configurations https://colab.research.google.com/github/recohut/notebook/blob/master/_notebooks/2021-06-25-kafka-spark-streaming-colab.ipynb