Lambda Architecture for Twitter real-time sentiment analysis

Lorenzo Agnolucci

Università degli Studi di Firenze
Dipartimento di Ingegneria dell'Informazione

Firenze, 20 Aprile 2019

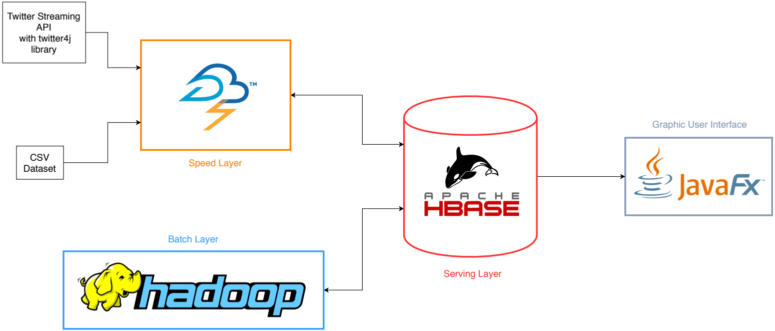
Outline

1. Introduction
2. Proposed approach
   - Sentiment classifier
   - Serving layer
   - Batch layer
   - Speed layer
   - GUI
3. Conclusions

Introduction

- Big Data requires to find ways to analyze a large amount of data
- Lambda Architecture is a particular approach composed by:
  - *batch layer* : applies batch-oriented technologies (like MapReduce) on a master database. It is effective but it has a high latency
  - *serving layer* : specialized distributed database that supports batch updates and random reads
  - *speed layer* : only looks at recent data and uses low-latency techniques to update real-time views. It compensate for the high latency of the batch layer
- Sentiment analysis is a type of data mining applied to Big Data with some useful applications

Proposed approach



- The main goal was not a perfect sentiment classification but the implementation of the architecture
- Speed layer is started first with the keywords as arguments. It creates the speed layer tables at the start of the execution
- Tweets of a dataset are added as if they belonged to the real-time stream to increase the number of tweets

Sentiment classifier

- Developed with *LingPipe* library
- Trained on 1.6 millions tweets [1]
- Classifies English text with 2 categories: positive and negative
- Decent 0.71 accuracy

[1] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. CS224N project report, Stanford, 1(12):2009, 2009. https://www.kaggle.com/kazanova/sentiment140

Serving layer

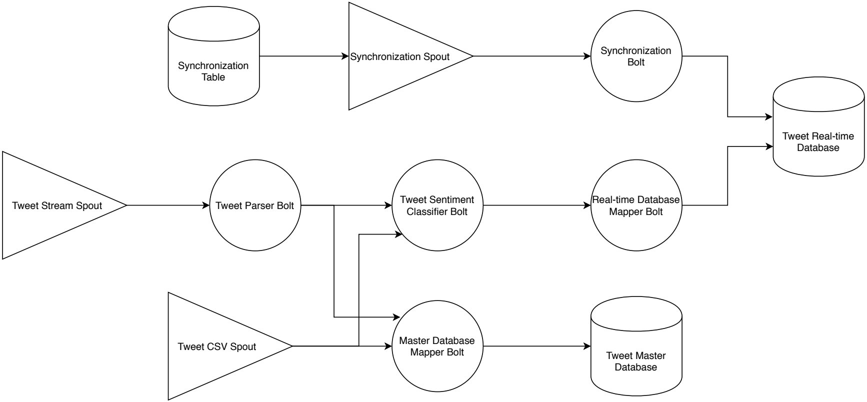
Based on Apache HBase and composed by 4 tables:

- *tweet master database*: master database of the Lambda Architecture
- *tweet real-time database*: stores the tweets on which the real-time view is based
- *batch view*: result of the batch processing
- *synchronization table*: contains the start and the end timestamps of the batch processing

Batch layer

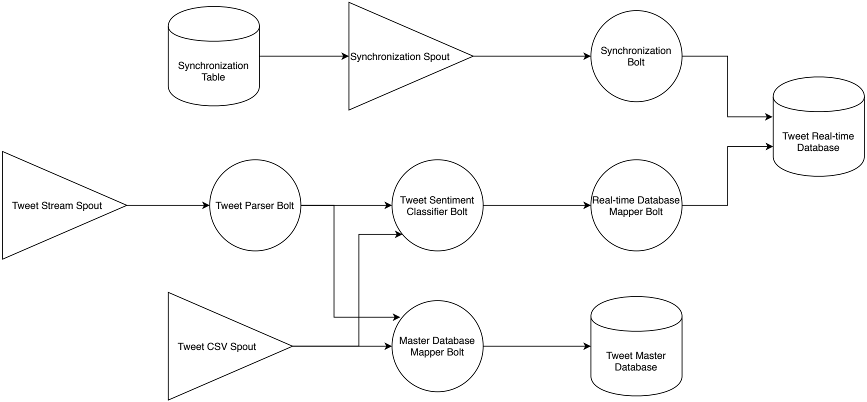
- Represented by Apache Hadoop
- Computes a MapReduce job on *tweet master database* in a infinite loop
- Writes its results from scratch in *batch view*
- Writes the start and the end timestamps of the computation in *synchronization table*
- Mapper takes a tweet in input and outputs a $< Keyword, Sentiment >$ tuple
- Reducer takes a tuple in input and increment the corresponding cell in *batch view*

Speed layer



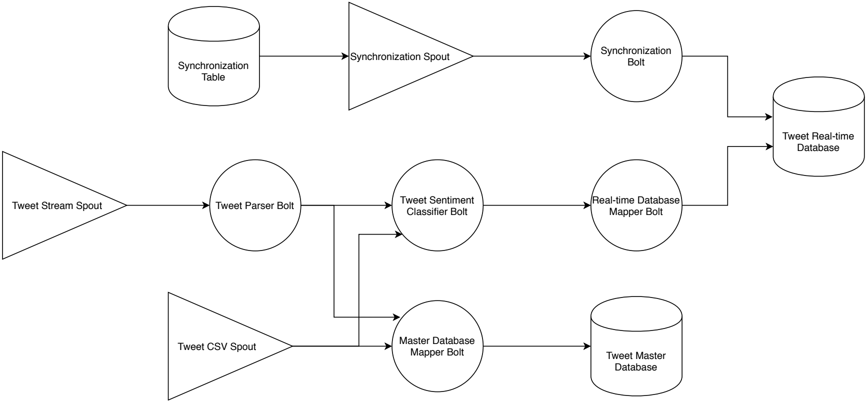
- *tweet stream spout*: gets a real-time stream of tweets with *Twitter4j* library and filters them
- *tweet parser bolt*: parse a tweet object to a tuple

Speed layer



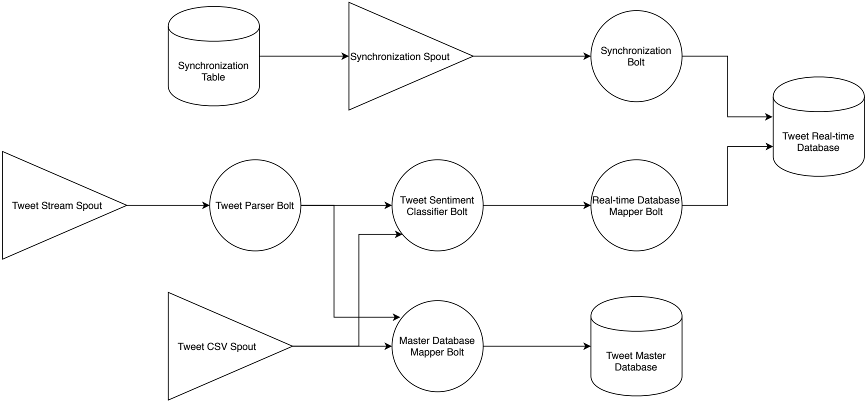
- *tweet CSV spout*: outputs a tuple for each tweet of the dataset
- *master database mapper bolt*: inserts tweets in *tweet master database*

Speed layer



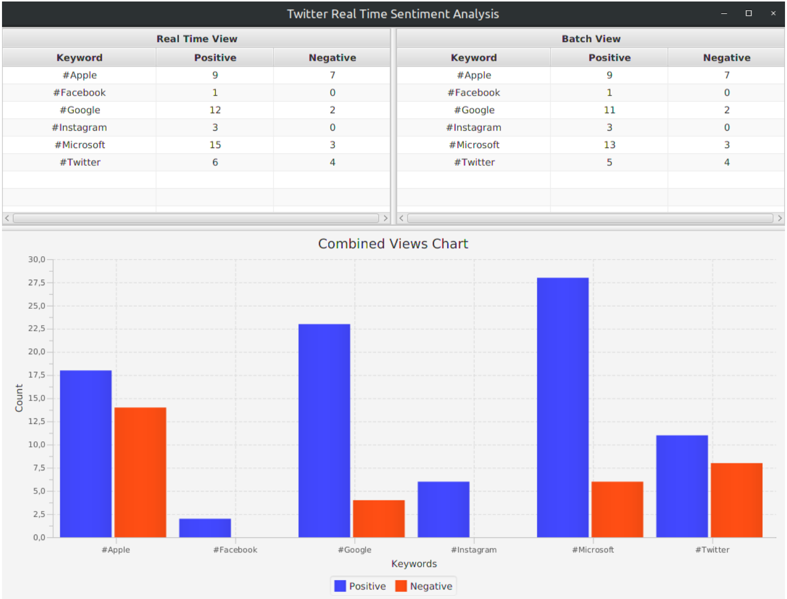
- *tweet sentiment classifier bolt*: classifies the sentiment of the tweet
- *real-time database mapper bolt*: inserts tuples in *tweet real-time database*

Speed layer



- *synchronization spout*: checks when batch processing ends
- *synchronization bolt*: deletes already processed tweets from *tweet real-time database*

GUI



Twitter Real Time Sentiment Analysis

Real Time View

Keyword	Positive	Negative
#Apple	9	7
#Facebook	1	0
#Google	12	2
#Instagram	3	0
#Microsoft	15	3
#Twitter	6	4

Batch View

Keyword	Positive	Negative
#Apple	9	7
#Facebook	1	0
#Google	11	2
#Instagram	3	0
#Microsoft	13	3
#Twitter	5	4

Conclusions

- It has been shown an implementation of a Lambda Architecture capable of getting sentiment analysis statistics of real-time tweets
- The GUI that was developed lets to visualize how the different parts of the architecture work together
- As a future development a neutral category could be added to the sentiment classifier