**NIST Big Data Public Working Group (NBD-PWG)**

**NBD-PWD-2015/Use-case-#3-Live-Twitter-Analysis-rr**

**Source: NBD-PWG**

**Status: Draft**

**Title: Possible Big Data Use Cases Implementation using NBDRA**

**Author: Afzal Godil (NIST), Wo Chang (NIST), Russell Reinsch (CFGIO)**

To support Version 2 development, six unique Big Data use cases (with publicly available datasets and analytic algorithms) are given for implementation using the NIST Big Data Reference Architecture (NBDRA). This document covers use case #3. We encourage NBD-PWG members to help implement them using NBDRA so that we can learn about the dataflow as well as their interactions between NBDRA key components.

**I. Introduction**

Social media for many people has become integral part of their daily life. Social media metrics are now considered parts of altmetrics, which are non-traditional metrics proposed as an alternative to more traditional metrics.

Background on Twitter: Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can post and read tweets, but general public can also read them. This is unlike Facebook, where social interactions are often private. Users access Twitter through the website interface, SMS, or mobile device app.

We will develop a program(s) for live Twitter Analysis based on Twitter's Search and Streaming APIs for sentiment analysis and for visualization of results (see Figure 1, M0399 which shows examples of different Twitter visualizations). We will also analyze and visualize the NIST Twitter network. We can track and statistically analyze the NIST mentions, followers, retweets, compare them to other National labs, and many more things. The analysis could help NIST measure and improve our effectiveness in engaging the public about our work and our outreach effort on Twitter.

We could develop the application based on Apache Storm, a distributed computation framework, which adds reliable real-time data processing capabilities to Apache Hadoop. It is fast, scalable, and reliable and can be programmed using a variety of programming languages (Python, Java, Scala). Its architecture consists of three primary node sets: Nimbus nodes, Zookeeper nodes, and Supervisor nodes.

Background on Sentiment Analysis: Sentiment analysis or opinion mining refers to the use of natural language processing and text analysis to identify and extract subjective information in source materials. Normally speaking, sentiment analysis aims to determine the attitude of a speaker or a writer with respect to some topic or the overall context of a communication.

Examples of words used for Sentiment analysis:

Positive: nice, awesome, cool, superb, etc.

Negative: bad, uninspired, expensive, disappointed, recommend others to avoid, etc.

Datasets: Live Twitter feed

Specific Questions: Develop tools for location based sentiment analysis of Twitter feed in real-time.

Possible Development Tools

Big-Data: Apache Storm, Apache HBase, Twitter's Search and Streaming APIs

Visualization tools: D3 Visualization, Tableau visualization.

Natural Language Processing Algorithms: Python Natural Language Toolkit (NLTK), AlchemyAPI Service

Languages: Java, Python, Scala, Javascript, JQuery

**II. Assumptions**

Data integrity needs to be addressed for each of the following components: OS, file system, DB, virtual machnes, middleware and applications.

**III. Generalized abstract of a typical Twitter sentiment analysis**

Acquire streaming data 🡪 storage 🡪 NRT analysis 🡪 retrieve for visualization and public interface

**IV. Flow scenarios**

**Main scenario**

TBD; set for version 2.

**Alternate flow scenario with Hortonworks HDF powered by Apache Nifi**

|  |  |  |
| --- | --- | --- |
| Role | Activity | Detail |
| Application provider | Collection | Setup extractor to fetch data (configure http, JMS, Kafka or Solr properties; specify endpoint, ). |
| Application provider | Collection | Setup destination, configure HDFS properties (location of config file, directory to write to, ). |
| Application provider | Collection | Configure connection (prioritization, backpressure threshold; routing decisions). |
| Application provider | Transformation | Data enrichment, convert csv to avro, avro to JSON, . |
| Framework provider | Processing | Run a single component or whole system. |
| Framework provider | Processing | Adjust properties and settings, tune data flow, add new processors. |
| Generalizing highlight or specific challenge |  |  |
| Reference information | Nifi: <https://nifi.apache.org/> |  |

**Alternate flow scenario with NLTK and Python**

|  |  |  |
| --- | --- | --- |
| Role | Functional Component | Detail |
| Orchestrator | Collection | Copy module source code from code provider and paste. |
| Orchestrator | Collection | Create new app: get consumer key [ckey], consumer secret [csecret], create [atoken] and [asecret] values from twitter apps. |
| Framework provider | Processing | Set confidence output |
| Application provider | Visualization | Copy graph updating code from code provider; import matplotlib; use ggplot for style. Pull data. |
|  | Integration (compute, storage, network, software) |  |
| Generalizing highlight or specific challenge | Streaming APIs use a lot of RAM and CPU. |  |
| Reference information | <https://www.youtube.com/watch?v=SB8ckgT8l9c> |  |

Further Reference:

Mining Youtube analysis using Python, including complete script: <http://www.analyticsvidhya.com/blog/2014/09/mining-youtube-python-social-media-analysis/>

<https://www.brighttalk.com/webcast/9573/179469>