**Spark Technical Test Documentation**

The included files contain the source code and output files for the Spark Technical Test.

**Results Output Files:**

Process reported **10,665** tweets were contained in the file.

user-ids.txt – Text file containing a list of all unique user IDs.

word-counts.csv – Text file containing unique words and counts.

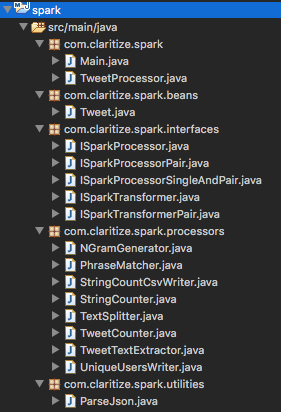
top-phrases.txt – Top N phrases matching the list (when N = 10).

n-grams.csv – N-grams and counts (when N = 5).

Source code is contained under the **source** sub-directory.

**Source Code:**

Below is a summary of the classes written to complete these tests:



**Package: com.claritize.spark**

**Main** – Class containing the main method that initiates the Tweet processor and specifies the Tweets and phrases file to be processed.

**TweetProcessor** – Manages the computation of the 5 tasks to be performed on these Tweets. Each task has it’s own method e.g. countTweets().

**Package: com.claritize.spark.beans**

**Tweet** – Class that represents a single Tweet. This is a limited implementation that only holds user IDs and Tweet text.

**Package: com.claritize.spark.interfaces**

This package contains multiple interfaces for Spark transformations (where a RDD is returned) and for processes (for a chain of transformations and an action).

Each interface represents different combinations of these. For example ISparkTransformerPair represents transformations that take a RDD and return a RDD Pair.

All interfaces use generics to allow any type of RDD to be processed.

**Package: com.claritize.spark.processors**

This package contains classes that are all implementations of one of the above interfaces.

**NGramGenerator** – creates a RDD of N-Grams from a supplied RDD of text (N is configurable). Used by the N-Gram task.

**PhraseMatcher** – matches a RDD of phrases to a RDD of word counts using a join. Matches are then sorted into a top N and stored to a file (N is configurable). Used by the phrase matching task.

**StringCountCsvWriter** – Creates a RDD of CSV rows and then writes them to a supplied directory. Used by the N-Gram and word count tasks for output.

**StringCounter** – counts the number of unique strings seen in the supplied RDD of strings. Returns a Pair RDD of strings and counts. Used by the N-Gram and word count tasks for output.

**TextSplitter** – splits a RDD of strings into a RDD for each word using a flatMap(). Used to create the word counts task. RDD is cached and re-used by the top phrases task.

**TweetCounter** – simple class that logs the number of Tweets seen in an RDD of Tweets. Used by the counting task.

**TweetTextExtractor** – extracts the Tweet text from each Tweet to return a RDD of strings. Cached and re-used by the word counts, top phrases and N-gram tasks.

**UniqueUsersWriter** - Creates a RDD of Tweet Ids from the RDD of Tweets. Then creates a RDD of unique User IDs and writes them to a supplied directory. Used by the unique users task.

**Package: com.claritize.spark.utilities**

**ParseJson** – Uses a FlatMap function to build a RDD of Tweet objects from a RDD of lines from the JSON file. Uses the Google GSON library.