Plagiarism Scan Report

Summary	
Report Genrated Date	23 Jul, 2018
Plagiarism Status	**100% Unique**
Total Words	281
Total Characters	2232
Any Ignore Url Used	

Content Checked For Plagiarism:

```
package twitterhashtag
import org.apache.spark._
import org.apache.spark.SparkContext._
import org.apache.spark.streaming._
import org.apache.spark.streaming.twitter._
import org.apache.spark.streaming.StreamingContext._
object PopularHashtags {

def setupLogging() = {
import org.apache.log4j.{Level, Logger}
val rootLogger = Logger.getRootLogger()
rootLogger.setLevel(Level.ERROR)
}

def setupTwitter() = {
import scala.io.Source

for (line <- Source.fromFile("./twitter.txt").getLines) {
val fields = line.split(" ")
if (fields.length == 2) {
System.setProperty("twitter4j.oauth." + fields(0), fields(1))
}
}
}
/** Our main function where the action happens */
def main(args: Array[String]) {
// Configure Twitter credentials using twitter.txt
setupTwitter()
// Set up a Spark streaming context named "PopularHashtags" that runs locally using
// all CPU cores and one-second batches of data
val ssc = new StreamingContext("local[*]", "PopularHashtags", Seconds(1))
// Get rid of log spam (should be called after the context is set up)
setupLogging()
// Create a DStream from Twitter using our streaming context
val tweets = TwitterUtils.createStream(ssc, None)
// Now extract the text of each status update into DStreams using map()
```

```
val statuses = tweets.map(status => status.getText())
val tweetwords = statuses.flatMap(tweetText => tweetText.split(" "))
// Now eliminate anything that's not a hashtag
val hashtags = tweetwords.filter(word => word.startsWith("#"))
val hashtagKeyValues = hashtags.map(hashtag => (hashtag, 1))
// Now count them up over a 5 minute window sliding every one second
val hashtagCounts = hashtagKeyValues.reduceByKeyAndWindow( (x,y) => x + y, (x,y) =>
x - y, Seconds(300), Seconds(1))
// You will often see this written in the following shorthand:
//val hashtagCounts = hashtagKeyValues.reduceByKeyAndWindow( _ + _, _ -_,
Seconds(300), Seconds(1))
// Sort the results by the count values
val sortedResults = hashtagCounts.transform(rdd => rdd.sortBy(x => x._2, false))
// Print the top 10
sortedResults.print
// Set a checkpoint directory, and kick it all off
// I could watch this all day!
ssc.checkpoint("/home/nikhil/Desktop/TwitterHashTag/checkpoints")
ssc.start()
ssc.awaitTermination()
}
}
```