**W251 – Homework 9**

Chula Watugala

Using 60 second sampling duration and 30min total duration with top 10 popular topics:

* The top *n* most frequently-occurring hash tags among all tweets during the sampling period (tweets containing these tags pick out 'popular' topics)

1. #5onthewall
2. #videoveranomtv
3. #travel
4. #fanarmyfaceoff
5. #choiceinternationalartist
6. #teenchoice
7. #littlemixgetweird
8. #lisemata10binata
9. #funnymoment
10. #nowplaying

* The account names of users who authored tweets on popular topics in the period
* pri5sos\_
* naioutside
* isabellmariie
* marilyn\_mald
* lavinraven
* httpmendestyles
* jonatounbroken
* itzabihail
* jasminedavies96
* ashiallx
* mizzlesbaby
* alecamzjergi
* cynthiajm95
* ari5hever
* milla\_silva96
* onlinervstore
* mytravelluck
* iprefersstyles
* fronia2
* just\_\_smg
* kondyaang13
* azii1991
* yalbalolo
* yalbalolo
* kondyaang13
* azii1991
* meredithganley
* lovethem\_\_
* tugbaysen
* srcncskn77
* natalyafateeva1
* reynarivest
* elmogollorevele
* radiosusans
* The account names of users who were mentioned in popular tweets *or* by popular people
* 5sos
* rt\_erdogan
* ahmet\_davutoglu
* kilicdarogluk
* amyschumer
* theellenshow

import org.apache.spark.storage.StorageLevel

import org.apache.spark.streaming.{Seconds, StreamingContext}

import org.apache.spark.streaming.StreamingContext.\_

import org.apache.spark.SparkContext.\_

import org.apache.spark.streaming.twitter.TwitterUtils

import org.apache.spark.SparkConf

object hw9 {

def main(args: Array[String]) {

// args(0) = sampling duration

// args(1) = total duration

// args(2) = number of popular topics

System.setProperty("twitter4j.oauth.consumerKey", "Ldd2233hMQvm9awpVH3kNF5ZC")

System.setProperty("twitter4j.oauth.consumerSecret", "8f01RUtGWNYdtWbmN399TipR8NApfaafqptDS4FiD9SmJLpdxD")

System.setProperty("twitter4j.oauth.accessToken", "86446655-gqB4EWTHkbrAH0MVQsJG0eQMpW9ZRzf17scF5x3qs")

System.setProperty("twitter4j.oauth.accessTokenSecret", "uqbF7Yc9zo1LWNrHTi4aW2V8W6uiRld0YvZGhe4wIgR3v")

val conf = new SparkConf().setAppName(getClass.getSimpleName)

val context = new StreamingContext(conf, Seconds(args(0).toLong))

val stream = TwitterUtils.createStream(context, None)

println(s"Sampling duration of ${args(0)} and total duration of ${args(1)}.")

// Drop @

val mentionR = """^@(\w{1,20})$""".r

// Keep #

val hashtagR = """^(#\w\*[A-Za-z\_]+\w\*)$""".r

// Extract hashtag and mentions

val parseTweet = (s: String) => s

.split("\\s+")

.foldLeft((Set.empty[String], Set.empty[String])){

case ((hashtags, mentions), token) => token match {

case hashtagR(t) => (hashtags + t, mentions)

case mentionR(t) => (hashtags, mentions + t)

case \_ => (hashtags, mentions)

}

}

val sampStream = stream

// map to (user, tweet text)

.map(tweet => (tweet.getUser.getScreenName.toLowerCase, tweet.getText.toLowerCase))

// map to (user, (hashtags, mentions)) using parseTweet

.mapValues(parseTweet(\_))

// map each hashtag to (hashtag, (user, mentions))

.flatMap{case (user, (hashtags, mentions)) => hashtags.map(hashtag => (hashtag, (user, mentions)))}

sampStream.window(Seconds(args(1).toLong)).foreachRDD(rdd => {

val top = rdd

// Aggregate hashtag user and mentions using count

.map{case (hashtag, (user, mentions)) => (hashtag, (1, Set(user), mentions))}

.reduceByKey((r, l) => (r.\_1 + l.\_1, r.\_2 ++ l.\_2, r.\_3 ++ l.\_3))

// sort by hashtag counts

.sortBy({case (\_, (count, \_, \_)) => count}, ascending = false)

// pick top X topics

.take(args(2).toInt)

println(s"Top ${args(2)} topics for sampling duration:")

top.foreach { case (hashtag, (count, users, mentions)) => {

println(s"Hashtag $hashtag ($count total):")

println(s" Users:")

println(" " + users.mkString("\n "))

println(s" Mentions:")

println(" " + (if (mentions.nonEmpty) mentions.mkString("\n ") else "[no mentions]"))

}

}

})

context.start()

context.awaitTermination()

context.stop()

}

}