2018W03.sentiment

```
println("Spark version " + sc.version)
// println("Scala version " + util.Properties.versionString)
```

FINISHED

```
Spark version 2.1.0
```

Took 1 min 8 sec. Last updated by anonymous at January 19 2018, 1:08:33 PM.

```
// Global variables
val dir_data = "data/dev-test-exploratory-analysis"
val f_sentipos = dir_data + "/sentiment_pos_norm"
val f_sentineg = dir_data + "/sentiment_neg_norm"
val f_lexemes = dir_data + "/lexemes_norm/*.csv"
val fcomm_clean = dir_data + "/224564804326967_facebook_comments_clean"
// End of Global variables
```

FINISHED

```
dir_data: String = data/dev-test-exploratory-analysis
f_sentipos: String = data/dev-test-exploratory-analysis/sentiment_pos_norm
f_sentineg: String = data/dev-test-exploratory-analysis/sentiment_neg_norm
f_lexemes: String = data/dev-test-exploratory-analysis/lexemes_norm/*.csv
fcomm_clean: String = data/dev-test-exploratory-analysis/224564804326967_facebook_comments_cl
ean
```

Took 1 min 7 sec. Last updated by anonymous at January 19 2018, 1:08:34 PM.

```
import org.apache.spark.rdd.RDD

// Read sentiment files
val sentPosDF = sc.textFile(f_sentipos)
val sentNegDF = sc.textFile(f_sentineg)

// Read lexemes to DataFrame
case class Lexeme(lemma: String, pragma: String)
val lexemesDF = sc.textFile(f_lexemes).map(_.split(",")).map(atr => Lexeme(atr(0).trim, atr(1

// Split lines of comment
val commentsDF = sc.textFile(fcomm_clean)
val commentsArrDF : RDD[Array[String]] = commentsDF.map(line => line.split("\\W"))
```

FINISHED

```
import org.apache.spark.rdd.RDD
sentPosDF: org.apache.spark.rdd.RDD[String] = data/dev-test-exploratory-analysis/sentiment_po
s_norm MapPartitionsRDD[1] at textFile at <console>:34
sentNegDF: org.apache.spark.rdd.RDD[String] = data/dev-test-exploratory-analysis/sentiment_ne
g_norm MapPartitionsRDD[3] at textFile at <console>:32
defined class Lexeme
lexemesDF: org.apache.spark.rdd.RDD[Lexeme] = MapPartitionsRDD[7] at map at <console>:34
commentsDF: org.apache.spark.rdd.RDD[String] = data/dev-test-exploratory-analysis/22456480432
6967_facebook_comments_clean MapPartitionsRDD[9] at textFile at <console>:34
commentsArrDF: org.apache.spark.rdd.RDD[Array[String]] = MapPartitionsRDD[10] at map at <cons
ole>:34
```

Took 4 sec. Last updated by anonymous at January 19 2018, 1:08:38 PM.

```
object LookupFun  extends java.io.Serializable {
  def lemword(word: String, mymap: Map[String,String]): String = {
      val res : String = mymap.get(word).mkString
      if (res.isEmpty) word else res
  }

  def arrword(arr: Array[String], mymap : Map[String,String]) : Array[String] = {
      arr.map(word => lemword(word, mymap))
  }
}
```

FINISHED

```
defined object LookupFun
```

Took 3 sec. Last updated by anonymous at January 19 2018, 1:08:38 PM.

```
println("Lexemes")
lexemesDF.take(3).foreach(println)
println("Postivie sentiments")
sentPosDF.take(3).foreach(println)
println("Negaivie sentiments")
sentNegDF.take(3).foreach(println)
```

FINISHED

```
Lexemes
Lexeme(cajchovat,cajchovat)
Lexeme(cajchovat,cajchujem)
Lexeme(cajchovat,cajchujes)
Postivie sentiments
right
ako
ma
Negaivie sentiments
o
polozeny
priemerny
```

Took 3 sec. Last updated by anonymous at January 19 2018, 1:08:41 PM.

```
// Lookup table
val lexemesMap : Map[String,String] = lexemesDF.map(r => (r.pragma, r.lemma)).collect.toMap
```

FINISHED

```
lexemesMap: Map[String,String] = Map(tiborov -> tibor, javskou -> javsky, karborundovemu -> k
arborundovy, startujem -> startovat, nezavrzdiac -> nezavrzdat, spustneme -> spustnut, dahoch
 -> dah, dzezista -> dzezista, previedol -> previest, brunejskeho -> brunejsky, strukturovane
j -> strukturovany, pribudol -> pribudnut, nepofrancuzstite -> nepofrancuzstit, najchybnejsej
 -> chybny, nefrfocte -> nefrfotat, neprikryvajte -> neprikryvat, niznorepasskej -> niznorepa
ssky, sexualny -> sexualny, clapotaveho -> clapotavy, cudzobaznejsim -> cudzobazny, neflochaj
uc -> neflochat, nepovarovala -> nepovarovat, hrabskych -> hrabsky, kvestormi -> kvestor, pre
tvaratelkach -> pretvaratelka, nesnupli -> nesnupnut, demagogicke -> demagogicka, brejkujes -
> brejkovat, nefuskarime -> nefuskarit, nezzelie -> n...
```

Took 7 sec. Last updated by anonymous at January 19 2018, 1:08:46 PM.

```
// Case Calsses (Models)
case class SentiProbM(negative: Double, neutral: Double, positive: Double)
case class SentiCountM(count: Long, negative: Long, positive: Long)

object SentiCount  extends java.io.Serializable {
    // Sentiments set contains words from array
    def wordsInSentiments (arr_w : Array[String], set_s : Set[String]) : Long = {
```

FINISHED

```
        var cnt : Long = 0
        arr_w.map{ w =>
            if (set_s.contains(w))
                cnt = cnt + 1
        }
        return cnt
    }
}

object SentiCompute  extends java.io.Serializable {
    def sentiToProb(C: SentiCountM): SentiProbM = {
        // return: Array(prob. negative, prob. neutral, prob. positive)
        val d_cnt: Double = C.count.toDouble
        val d_neg: Double = C.negative.toDouble
        val d_pos: Double = C.positive.toDouble
        val pneg: Double = d_neg/d_cnt    // Probability of negative sentiment
        val ppos: Double = d_pos/d_cnt    // --"-- positive sentiment
        val pneu: Double = (d_cnt - (d_neg + d_pos))/d_cnt // --"-- neutral sentiment
        return SentiProbM(pneg, pneu, ppos)
    }

    def maxIdx(arr: Array[Double]) : Int = {
        val result = arr.foldLeft(-1,Double.MinValue,0) {
            case ((maxIndex, maxValue, currentIndex), currentValue) =>
                if(currentValue > maxValue) (currentIndex,currentValue,currentIndex+1)
                else (maxIndex,maxValue,currentIndex+1)
            }
        result._1
    }

    def sentiLabel(P: SentiProbM) : Int = {
        if (P.positive == P.negative) {
             return 0 // Neutral
        } else {
            if (P.positive > P.negative) {
                return 1 // Positive
            } else {
                return -1 // Negative
            }
        }
    }
}
```

```
defined class SentiProbM
defined class SentiCountM
defined object SentiCount
defined object SentiCompute
```

Took 1 sec. Last updated by anonymous at January 19 2018, 1:08:49 PM. (outdated)

```
val tstfnComment = commentsArrDF.take(1)(0)
val sentPosSet = sentPosNormForm.toSet
val sentNegSet = sentNegNormForm.toSet

val cntPos = SentiCount.wordsInSentiments(tstfnComment, sentPosSet)
val cntNeg = SentiCount.wordsInSentiments(tstfnComment, sentNegSet)
val cntWord = tstfnComment.length
val cntNeutr = cntWord - (cntPos + cntNeg)
println("Number of words " + cntWord)
println("Number of + words " + cntPos)
println("Number of - words " + cntNeg)
println("Number of 0 words " + cntNeutr)
```

FINISHED

```
tstfnComment: Array[String] = Array(nieco, uzasne, vyklepky, staly, 9, kcs, 30, ks, teraz,
 dva, ta, cena, byt, vidiet, mama, dobre, bluznit, sudruh, birmovany, byt, nacisto, osprost
eny)
```

```
sentPosSet: scala.collection.immutable.Set[String] = Set(klasicky, pamatny, drahokam, veden
ie, znak, ocenovat, zivit, vynikajuci, ovladat, vyhovujuci, autonomny, praca, renesancia, p
ravdivy, posobivy, asociativny, pocitit, obnovenie, modny, vedeny, uprimnost, nestranny, la
skavy, doplnkovy, hrdinsky, vazny, podporujuci, cisto, posadnutost, zavazok, citlivy, raj,
 horlivy, zmiernit, harmonium, bezpochyby, odtlacok, nebesky, support, odporuceny, rurka, h
ladko, moderna, platit, doplnat, talentovany, ziara, konstruktivny, trend, dokonaly, pozito
k, osvietenstvo, bohatost, lead, instrumentalny, win, zjednodusit, revival, tazky, povznese
nie, svetly, lesknut, prestizny, mudro, veltrh, schopny, nadherny, zastanca, prisposobivy,
 opravneny, splnat, chciet, jasny, hlboko, pycha, obrodenie, fantazia, ...sentNegSet: scal
a.collection.immutable.Set[String] = Set(breaks, trest, predmet, lono, prisny, preruseny, c
hybat, explodovat, odrazit, kut, eliminacia, samovrazda, kostra, zlomit, vona, izolovat, ne
dostatok, zakazany, vyhnat, trhlina, ukradnut, beg, zrucanina, neposlusnost, opustit, preru
senie, zastrasit, dozadu, pohltit, choroba, zmateny, rue, neaktivny, tlmic, prenikat, poraz
```

Took 4 sec. Last updated by anonymous at January 19 2018, 1:08:52 PM.

FINISHED

```
// Example: Usage of classes SentiCount & SentiCompute
val sentPosSet = sentPosNormForm.toSet
val sentNegSet = sentNegNormForm.toSet

val tstfnComment = commentsArrDF.take(10).map{arr =>
    val cntPos = SentiCount.wordsInSentiments(arr, sentPosSet)
    val cntNeg = SentiCount.wordsInSentiments(arr, sentNegSet)
    val cntWord = arr.length
    val S = SentiCountM(cntWord, cntNeg, cntPos)
    println(S)
    SentiCompute.sentiToProb(S)
}
```

```
sentPosSet: scala.collection.immutable.Set[String] = Set(klasicky, pamatny, drahokam, veden
ie, znak, ocenovat, zivit, vynikajuci, ovladat, vyhovujuci, autonomny, praca, renesancia, p
ravdivy, posobivy, asociativny, pocitit, obnovenie, modny, vedeny, uprimnost, nestranny, la
skavy, doplnkovy, hrdinsky, vazny, podporujuci, cisto, posadnutost, zavazok, citlivy, raj,
 horlivy, zmiernit, harmonium, bezpochyby, odtlacok, nebesky, support, odporuceny, rurka, h
ladko, moderna, platit, doplnat, talentovany, ziara, konstruktivny, trend, dokonaly, pozito
k, osvietenstvo, bohatost, lead, instrumentalny, win, zjednodusit, revival, tazky, povznese
nie, svetly, lesknut, prestizny, mudro, veltrh, schopny, nadherny, zastanca, prisposobivy,
 opravneny, splnat, chciet, jasny, hlboko, pycha, obrodenie, fantazia, ...sentNegSet: scal
a.collection.immutable.Set[String] = Set(breaks, trest, predmet, lono, prisny, preruseny, c
hybat, explodovat, odrazit, kut, eliminacia, samovrazda, kostra, zlomit, vona, izolovat, ne
dostatok, zakazany, vyhnat, trhlina, ukradnut, beg, zrucanina, neposlusnost, opustit, preru
senie, zastrasit, dozadu, pohltit, choroba, zmateny, rue, neaktivny, tlmic, prenikat, poraz
eny, marit, nerozhodnost, nizky, prisera, nestaly, vytvoreny, nespolahlivy, ospravedlnenie,
 ulozit, potopeny, namietka, prenasledovanie, sporny, tazko, zmiesany, zarobit, duty, chybn
y, vypalenie, pristav, vladnut, povest, diskriminacia, nahrubo, poskodenie, krutost, treni
e, vymysleny, hroziace, tresk, otrocit, vypovedat, blokada, prudko, zastavenie, hmla, nepac
it, zraneny, tazky, opuch, kazat, patentovany, kos, pr...SentiCountM(22,0,2)
```

Took 1 sec. Last updated by anonymous at January 19 2018, 1:08:53 PM.

FINISHED

```
val sentiComment = commentsArrDF.collect.toArray.map{arr =>
    val cntPos = SentiCount.wordsInSentiments(arr, sentPosSet)
    val cntNeg = SentiCount.wordsInSentiments(arr, sentNegSet)
    val cntWord = arr.length
    val S = SentiCountM(cntWord, cntNeg, cntPos)
    SentiCompute.sentiToProb(S)
}
```

```
sentiComment: Array[SentiProbM] = Array(SentiProbM(0.0,0.9090909090909091,0.0909090909090909
1), SentiProbM(0.045454545454545456,0.9090909090909091,0.045454545454545456), SentiProbM(0.22
22222222222222,0.5555555555555556,0.2222222222222222), SentiProbM(0.0,0.8461538461538461,0.15
384615384615385), SentiProbM(0.1111111111111111,0.7777777777777778,0.1111111111111111), Senti
ProbM(0.3333333333333333,0.6666666666666666,0.0), SentiProbM(0.0,1.0,0.0), SentiProbM(0.0,1.
0,0.0), SentiProbM(0.0,1.0,0.0), SentiProbM(0.16666666666666666,0.6666666666666666,0.16666666
666666666), SentiProbM(0.0,1.0,0.0), SentiProbM(0.0,0.7857142857142857,0.21428571428571427),
 SentiProbM(0.16666666666666666,0.8333333333333334,0.0), SentiProbM(0.0,0.875,0.125), SentiPr
obM(0.2777777777777778,0.6111111111111112,0.111111111111...
```

Took 1 sec. Last updated by anonymous at January 19 2018, 1:08:54 PM.

FINISHED

```
// Create index and zip comment lines with sentiment probabilities
case class SentiRow (comment: String, index: Long, negative: Double, neutral: Double, positiv
val commentsSenti = commentsDF.zipWithIndex.map{t =>
    val line = t._1
    val idx = t._2
    val S: SentiProbM = sentiComment(idx.toInt)
    SentiRow(line, idx, S.negative, S.neutral, S.positive)
}
val commentsSentiDF = commentsSenti.toDF
commentsSentiDF.show()
```

```
defined class SentiRow
commentsSenti: org.apache.spark.rdd.RDD[SentiRow] = MapPartitionsRDD[14] at map at <console
>:82
commentsSentiDF: org.apache.spark.sql.DataFrame = [comment: string, index: bigint ... 3 mor
e fields]
+--------------------+-----+--------------------+------------------+--------------------+
|             comment|index|            negative|           neutral|            positive|
+--------------------+-----+--------------------+------------------+--------------------+
|nieco uzasne vykl...|    0|                 0.0|0.9090909090909091| 0.09090909090909091|
|byt zaujimavy cr ...|    1|0.045454545454545456|0.9090909090909091|0.045454545454545456|
|naozaj pomaly vel...|    2|  0.2222222222222222|0.5555555555555556|  0.2222222222222222|
|ano zvysovanie by...|    3|                 0.0|0.8461538461538461| 0.15384615384615385|
|julius holz zauji...|    4|  0.1111111111111111|0.7777777777777778|  0.1111111111111111|
|     konecna byt cas|    5|  0.3333333333333333|0.6666666666666666|                 0.0|
|                null|    6|                 0.0|               1.0|                 0.0|
|                null|    7|                 0.0|               1.0|                 0.0|
|                   0|    8|                 0.0|               1.0|                 0.0|
|jak kriz kriz stv...|    9| 0.16666666666666666|0.6666666666666666| 0.16666666666666666|
```

Took 16 sec. Last updated by anonymous at January 19 2018, 1:10:49 PM.

FINISHED

```
/* Save data to Parquet file */
import spark.implicits._

val dir_data = "data/dev-test-exploratory-analysis"
val f_sentdat = dir_data + "/sentiment-prob.parquet"
println("Write sentiment probabilities to parquet file " + f_sentdat)
val sentiCommentParquet = sc.parallelize(sentiComment).toDF
sentiCommentParquet.write.mode("overwrite").parquet(f_sentdat)
```

```
import spark.implicits._
dir_data: String = data/dev-test-exploratory-analysis
f_sentdat: String = data/dev-test-exploratory-analysis/sentiment-prob.parquet
Write sentiment probabilities to parquet file data/dev-test-exploratory-analysis/sentiment-pr
ob.parquet
sentiCommentParquet: org.apache.spark.sql.DataFrame = [negative: double, neutral: double ...
 1 more field]
```

Took 4 sec. Last updated by anonymous at January 19 2018, 1:11:07 PM. (outdated)

```
senticommentParquet.describe("negative", "neutral", "positive").show()
```

FINISHED

```
+-------+-------------------+-------------------+-------------------+
|summary|           negative|            neutral|           positive|
+-------+-------------------+-------------------+-------------------+
|  count|               1015|               1015|               1015|
|   mean|0.04824007631624003| 0.8661415027215451|0.08561842096221492|
| stddev|0.09573055129669215|0.17441492363496486|0.12924100532277838|
|    min|                0.0|               -1.0|                0.0|
|    max|                1.0|                1.0|                1.0|
+-------+-------------------+-------------------+-------------------+
```

Took 1 sec. Last updated by anonymous at January 19 2018, 1:11:09 PM.

```
// Some errors.
sentiCommentParquet.filter($"neutral" < 0.0).show()
```

FINISHED

```
+------------------+-------------------+------------------+
|          negative|            neutral|          positive|
+------------------+-------------------+------------------+
|0.6666666666666666|-0.3333333333333333|0.6666666666666666|
|               1.0|               -1.0|               1.0|
+------------------+-------------------+------------------+
```

Took 1 sec. Last updated by anonymous at January 19 2018, 1:37:32 PM. (outdated)

```
// val arrQ = Array(0.01, 0.03, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.97, 0.99)
// Q_0.25, 25% of all data is in quantil interval 0.25, (0.0, 0.25)
// Q_0.5,  50% of all data is in quantil interval 0.5,  (0.0, 0.50)

val arrQ = Array(0.25, 0.5, 0.75)
println("Quantiles for negative sentiment words")
val qNeg = sentiCommentParquet.stat.approxQuantile("negative",
           arrQ ,0.0)

println("Quantiles for neutral sentiment words")
val qNeu = sentiCommentParquet.stat.approxQuantile("neutral",
           arrQ ,0.0)

println("Quantiles for positive sentiment words")
val qPos = sentiCommentParquet.stat.approxQuantile("positive",
           arrQ ,0.0)


val qZip = ((qNeg, qNeu, qPos).zipped.toArray zip arrQ) map { case ((av,bv,cv), dv) => (av,bv
// (a, b, c).zipped.toList
val qDF = sc.parallelize(qZip).toDF("negative", "neutral", "positive", "quantil")
qDF.show()
```

FINISHED

```
arrQ: Array[Double] = Array(0.25, 0.5, 0.75)
Quantiles for negative sentiment words
qNeg: Array[Double] = Array(0.0, 0.0, 0.07692307692307693)
Quantiles for neutral sentiment words
qNeu: Array[Double] = Array(0.8, 0.9, 1.0)
Quantiles for positive sentiment words
qPos: Array[Double] = Array(0.0, 0.029411764705882353, 0.13333333333333333)
```

```
qZip: Array[(Double, Double, Double, Double)] = Array((0.0,0.8,0.0,0.25), (0.0,0.9,0.029411
764705882353,0.5), (0.07692307692307693,1.0,0.13333333333333333,0.75))
qDF: org.apache.spark.sql.DataFrame = [negative: double, neutral: double ... 2 more fields]
+-------------------+-------+--------------------+-------+
|           negative|neutral|            positive|quantil|
+-------------------+-------+--------------------+-------+
|                0.0|    0.8|                 0.0|   0.25|
|                0.0|    0.9|0.029411764705882353|    0.5|
|0.07692307692307693|    1.0| 0.13333333333333333|   0.75|
```

Took 3 sec. Last updated by anonymous at January 19 2018, 2:08:52 PM. (outdated)

READY