

# 2018W03.graph

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

// Global variables
val dir_data = "data/dev-test-exploratory-analysis"
val fngram = dir_data + "/224564804326967_facebook_comments_ngram"
// End of Global variables

// Read text file
val ngramsDF = spark.read
  .format("csv").load(fngram).toDF("ngram", "ngram_frequency")
```

FINISHED

Spark version 2.1.0

dir\_data: String = data/dev-test-exploratory-analysis

fngram: String = data/dev-test-exploratory-analysis/224564804326967\_facebook\_comments\_ngram

ngramsDF: org.apache.spark.sql.DataFrame = [ngram: string, ngram\_frequency: string]

Took 2 sec. Last updated by anonymous at January 14 2018, 6:21:48 PM.

```
ngramsDF.filter($"ngram_frequency" > 10).show()
```

FINISHED

```
+-----+-----+
|      ngram|ngram_frequency|
+-----+-----+
|lidl slovensko|      25.0|
|      byt on|      16.0|
|  super cena|      15.0|
|  byt kupit|      13.0|
|  toto byt|      12.0|
+-----+-----+
```

Took 2 sec. Last updated by anonymous at January 14 2018, 6:21:49 PM.

```
// Prepare vertex data
```

FINISHED

```
// val zipUDF = udf { (terms: Seq[String], probabilities: Seq[Double]) => terms.zip(probabiliti
val arrUDF = udf {(v: String) => Array(v.mkString)}
val vertexArr = vertexDF
  .withColumn("v0_arr", arrUDF(col("v0")))
  .withColumn("v1_arr", arrUDF(col("v1")))
```

arrUDF: org.apache.spark.sql.expressions.UserDefinedFunction = UserDefinedFunction(<function1>, ArrayType(StringType, true), Some(List(StringType)))

vertexArr: org.apache.spark.sql.DataFrame = [v0: string, v1: string ... 3 more fields]

Took 12 sec. Last updated by anonymous at January 14 2018, 6:22:13 PM.

```
val vertexMap = vertexTmp.select(
  col("v0WithIndex._1").as("v0term"),
  col("v1WithIndex._1").as("v1term"),
  col("v0WithIndex._2").as("v0num"),
```

FINISHED

```
col("v1WithIndex._2").as("v1num"),
col("weight"))
.orderBy(desc("weight"))
```

```
vertexMap: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [v0term: string, v1term:
string ... 3 more fields]
```

Took 1 sec. Last updated by anonymous at January 14 2018, 6:59:15 PM.

```
vertexTmp.select(
  col("v0WithIndex._1").as("v0term"),
  col("v1WithIndex._1").as("v1term"),
  col("v0WithIndex._2").as("v0num"),
  col("v1WithIndex._2").as("v1num"),
  col("weight"))
.orderBy(desc("weight")).show()
```

FINISHED

```
+-----+-----+-----+-----+-----+
| v0term|v1term|v0num|v1num|weight|
+-----+-----+-----+-----+-----+
| prosit|  vas|   56|   12|   9.0|
|      u|  vas|   27|   12|   9.0|
| dobry|  den|   79|  137|   9.0|
| kriz| nebyt|    7|   17|   8.0|
| byt| super|    0|   66|   8.0|
| byt|  vas|    0|   12|   7.0|
| zly| byt|   95|    0|   7.0|
| aky| byt|   40|    0|   7.0|
| vajicko| byt|    8|    0|   7.0|
| mily| lidl|  211|    6|   7.0|
| byt| tam|    0|   23|   6.0|
| byt| taky|    0|   29|   6.0|
| kupit| byt|   10|    0|   6.0|
| cena| vajce|    1|    4|   6.0|
| ktorvi| hvti|   38|    0|   6.0|
```

Took 10 sec. Last updated by anonymous at January 14 2018, 6:56:04 PM.

```
// Compute the highest degree vertex
// Define a reduce operation to compute the highest degree vertex
def max(a: (VertexId, Int), b: (VertexId, Int)): (VertexId, Int) = {
  if (a._2 > b._2) a else b
}
val maxInDegree: (VertexId, Int) = graphDat.inDegrees.reduce(max)
maxInDegree
val maxOutDegree: (VertexId, Int) = graphDat.outDegrees.reduce(max)
maxOutDegree
val maxDegrees: (VertexId, Int) = graphDat.degrees.reduce(max)
maxDegrees
```

FINISHED

```
max: (a: (org.apache.spark.graphx.VertexId, Int), b: (org.apache.spark.graphx.VertexId, Int))
(org.apache.spark.graphx.VertexId, Int)
maxInDegree: (org.apache.spark.graphx.VertexId, Int) = (0,285)
res270: (org.apache.spark.graphx.VertexId, Int) = (0,285)
maxOutDegree: (org.apache.spark.graphx.VertexId, Int) = (0,286)
res271: (org.apache.spark.graphx.VertexId, Int) = (0,286)
maxDegrees: (org.apache.spark.graphx.VertexId, Int) = (0,571)
res272: (org.apache.spark.graphx.VertexId, Int) = (0,571)
```

Took 2 sec. Last updated by anonymous at January 14 2018, 8:00:28 PM. (outdated)

FINISHED

```
vertexMap.filter("v1num == 285")  
  .show()
```

```
+-----+-----+-----+-----+-----+  
|  v0term|v1term|v0num|v1num|weight|  
+-----+-----+-----+-----+-----+  
|nekupovat|  pojd|  158|  285|   1.0|  
|  vajicko|  pojd|    8|  285|   1.0|  
|    cena|  pojd|    1|  285|   1.0|  
+-----+-----+-----+-----+-----+
```

Took 23 sec. Last updated by anonymous at January 15 2018, 4:32:35 PM.

FINISHED

```
vertexMap.filter("v1num == 286")  
  .show()
```

```
+-----+-----+-----+-----+-----+  
| v0term|v1term|v0num|v1num|weight|  
+-----+-----+-----+-----+-----+  
|hranica| najst|  433|  286|   1.0|  
|   vas| najst|   12|  286|   1.0|  
| nieco| najst|  154|  286|   1.0|  
|   kde| najst|   96|  286|   1.0|  
+-----+-----+-----+-----+-----+
```

Took 24 sec. Last updated by anonymous at January 15 2018, 4:32:24 PM.

FINISHED

```
vertexMap.filter("v0num == 571")  
  .show()
```

```
+-----+-----+-----+-----+-----+  
|v0term| v1term|v0num|v1num|weight|  
+-----+-----+-----+-----+-----+  
| jedlo|   kde|  571|   96|   1.0|  
| jedlo|auticko| 571|  229|   1.0|  
+-----+-----+-----+-----+-----+
```

Took 13 sec. Last updated by anonymous at January 15 2018, 4:32:11 PM.

FINISHED

```
vertexMap.filter("v0num == 0").show()  
vertexMap.filter("v1num == 0").show()
```

```
+-----+-----+-----+-----+-----+  
|v0term|  v1term|v0num|v1num|weight|  
+-----+-----+-----+-----+-----+  
|  byt|   super|    0|   66|   8.0|  
|  byt|    vas|    0|   12|   7.0|  
|  byt|   taky|    0|   29|   6.0|  
|  byt|    tam|    0|   23|   6.0|  
|  byt|vyjadrit|    0|  216|   6.0|  
|  byt|  chciet|    0|   30|   6.0|  
|  byt|   vzdy|    0|  134|   6.0|  
|  byt|    u|    0|   27|   5.0|  
|  byt| pravda|    0|   84|   5.0|
```

|  |               |   |     |     |
|--|---------------|---|-----|-----|
|  | byt slovensky | 0 | 15  | 4.0 |
|  | byt zly       | 0 | 95  | 4.0 |
|  | byt este      | 0 | 11  | 4.0 |
|  | byt fakt      | 0 | 118 | 4.0 |
|  | bvt mvsliet   | 0 | 109 | 4.0 |

Took 27 sec. Last updated by anonymous at January 15 2018, 4:45:26 PM.

READY