Paul 7 - enrich do...

```
%pyspark                                                                          FINISHED

# Zeppelin notebook to enrich domain summaries (from Paul 5) with examples (from Paul 6
# and topic metadata (from Tom 1)
# PJ - 9 November 2017

import boto
from pyspark.sql.types import *

# Load Domain Summaries DF in Gzip files (from Paul 5)
loadURL="s3://billsdata.net/CommonCrawl/domain_summaries6/" # Split into 20 files (as o
codec="org.apache.hadoop.io.compress.GzipCodec"
summary_df=spark.read.format('com.databricks.spark.csv').options(header='true', codec=c
summary_df.show(3)
summary_df.cache()
summary_df.rdd.getNumPartitions()
```

```
+--------------+--------+------------+----------+-----------+-------------+------+----
--+
|payLevelDomain|numHosts|pldIsHostFlag|pldLinksIn|pldLinksOut|wasCrawledFlag|hcRank|prRa
nk|
+--------------+--------+------------+----------+-----------+-------------+------+----
--+
|         37.ac|       1|         true|         1|       null|         false|49.503|47.7
23|
|      equal.ac|       1|         true|         3|          3|          true|69.571|68.0
98|
|      happy.ac|       1|         true|         2|          6|          true|19.030|69.9
99|
+--------------+--------+------------+----------+-----------+-------------+------+----
--+
only showing top 3 rows
20
```

```
%pyspark                                                                          FINISHED

# Load examples dataframe (from Paul 6)
examplesURI="s3://billsdata.net/CommonCrawl/domain_examples4/"
example_df=spark.read.load(examplesURI)
example_df.show(10)
example_df.cache()
example_df.rdd.getNumPartitions()
```

```
+--------------------+--------------------+--------------------+--------------------+
|                 PLD|        exampleHosts|   exampleInLinkPlds|  exampleOutLinkPlds|
+--------------------+--------------------+--------------------+--------------------+
|0----------------...|[0----------------...|         [nomina.ru]|                null|
|            0-0la.com|         [0-0la.com]|[jessicawilson.co...|[wordpress.org, g...|
|            0-3-6.com|         [0-3-6.com]|[3d114.com, menok...|                null|
|            0-3ani.ro|         [0-3ani.ro]|[cere.ro, adedir....|                null|
|            0-5-1.com|         [0-5-1.com]|    [allthecom.info]|                null|
|        0-60times.net|     [0-60times.net]|[lodekka.com, kev...|                null|
|             0-744.cn|          [0-744.cn]|     [wordpress.com]|                null|
|0-ads-free-web-pa...|[0-ads-free-web-p...|[list-of-domains....|                null|
|        0-artlove.net|     [0-artlove.net]|[list-of-domains....|                null|
|  0-clubpenguin-0.tk|[0-clubpenguin-0.tk]|  [similarsites.com]|                null|
+--------------------+--------------------+--------------------+--------------------+
only showing top 10 rows
256
```

```
%pyspark                                                                    FINISHED

# Remove square brackets from list output and remove Nulls
from pyspark.sql.functions import udf
def remove_brackets(egs):
    return str(egs).replace('[', '').replace(']', '')
print(remove_brackets("[bla, bla]"))
udf_remove_brackets = udf(remove_brackets, StringType())

example_df2=example_df.filter(example_df.PLD.isNotNull()).withColumn("tmp", udf_remove_
    ,"exampleOutLinkPlds")
example_df3=example_df2.withColumn("tmp", udf_remove_brackets("exampleInLinkPlds")).drop
example_df4=example_df3.withColumn("tmp", udf_remove_brackets("exampleHosts")).drop("ex
    ,"exampleOutLinkPlds")
example_df4.show(10)
example_df4.count()
```

```
bla, bla
+--------------------+--------------------+--------------------+--------------------+
|                 PLD|        exampleHosts|   exampleInLinkPlds|  exampleOutLinkPlds|
+--------------------+--------------------+--------------------+--------------------+
|0----------------...|0----------------...|           nomina.ru|                None|
|            0-0la.com|           0-0la.com|jessicawilson.co....|wordpress.org, gm...|
|            0-3-6.com|           0-3-6.com|3d114.com, menok....|                None|
|            0-3ani.ro|           0-3ani.ro|cere.ro, adedir.info|                None|
|            0-5-1.com|           0-5-1.com|      allthecom.info|                None|
|        0-60times.net|       0-60times.net|lodekka.com, kevi...|                None|
|             0-744.cn|            0-744.cn|       wordpress.com|                None|
|0-ads-free-web-pa...|0-ads-free-web-pa...| list-of-domains.org|                None|
|        0-artlove.net|       0-artlove.net| list-of-domains.org|                None|
|  0-clubpenguin-0.tk|  0-clubpenguin-0.tk|    similarsites.com|                None|
+--------------------+--------------------+--------------------+--------------------+
only showing top 10 rows
90839924
```

```
%pyspark                                                                    FINISHED

# Join with Original summaries
example_df.unpersist()
example_summary_df=summary_df.join(example_df4, summary_df.payLevelDomain==example_df4.l
summary_df.unpersist()
example_summary_df.dropDuplicates().sort("numHosts",ascending=False).show(100)
example_summary_df.count()
```

```
+--------------------+--------+-------------+----------+-----------+--------------+-----
-+------+--------------------+--------------------+--------------------+
|      payLevelDomain|numHosts|pldIsHostFlag|pldLinksIn|pldLinksOut|wasCrawledFlag|hcRan
k|prRank|        exampleHosts|   exampleInLinkPlds|  exampleOutLinkPlds|
+--------------------+--------+-------------+----------+-----------+--------------+-----
-+------+--------------------+--------------------+--------------------+
|        hotsited.com|   99995|         true|        29|          8|          true|66.63
8|61.568|hotsited.com, arc...|rdlj.ir, digicamh...|google.com, googl...|
|  unknownsecret.info|    9997|         true|        20|       null|         false|88.45
1|80.312|unknownsecret.inf...|list-of-domains.o...|                None|
|         b2csoez.com|     999|         true|        12|          7|          true|95.65
7|74.462|b2csoez.com, 007s...|blogspot.com, blo...|yahoo.com, f701.c...|
|            557b.com|     999|         true|       108|         70|          true|93.51
2|96.842|557b.com, 110001....|5ccs.com, ek59.co...|557h.com, 9ttu.co...|
|          dtyyt1.com|     999|         true|        12|          3|          true|89.67
3|67.896|dtyyt1.com, a.dty...|blogspot.com, blo...|ticrf.org.tw, eet...|
|          cbm665.com|     999|         true|        50|         19|          true|89.65
2|80.355|cbm665.com, a.cbm...|tyuqw59.com, hh19...|qymm088.com, hkk5...|
```

```
%pyspark                                                                    FINISHED

# Save Example Summaries as GZIP files, approx 100MB each.
#outURI="s3://billsdata.net/CommonCrawl/domain_summaries_withexamples4/"
#codec="org.apache.hadoop.io.compress.GzipCodec"
#example_summary_df.coalesce(40).write.format('com.databricks.spark.csv').options(header
print("SAVE HERE FOR ONLY EXAMPLE SUMMARIES")
```

```
SAVE HERE FOR ONLY EXAMPLE SUMMARIES
```

```
%pyspark                                                                    FINISHED

# Load topic labels (from Tom 1)
loadURL="s3://billsdata.net/CommonCrawl/topic_model_2048_files/cc_index_page_topic_label
topic_df=spark.read.load(loadURL)
topic_df.show(3)
print(topic_df.count())
topic_df.cache()
topic_df.rdd.getNumPartitions()
```

```
+--------------------+--------------------+--------------------+-------------------+----
----------------+--------------------+--------------------+--------------------+
|                host|                 url|              topic1|             score1|    
topic2|              score2|              topic3|              score3|
+--------------------+--------------------+--------------------+-------------------+----
----------------+--------------------+--------------------+--------------------+
|  astro.uchicago.edu|http://astro.uchi...|news_article_new_...|0.31914721112987016|said
_like_one_lov...| 0.27035210388505515|university_resear...| 0.14147791332687423|
|blog.greenmountai...|http://blog.green...|said_like_one_lov...| 0.2992614679052542|hote
ls_hotel_reso...| 0.21444496753319794|views_january_day...| 0.17792143718874276|
|        cleotube.com|http://cleotube.c...|porn_sex_tube_vid...| 0.9935485634990251|isla
nds_republic_...|0.003696503953861141|page_wiki_add_com...|2.863101549746408...|
+--------------------+--------------------+--------------------+-------------------+----
----------------+--------------------+--------------------+--------------------+
only showing top 3 rows
105744
147
```

```
%pyspark                                                                        FINISHED

# From Paul 5.

# Load in an uncompressed, partitioned format, for fast reading in the future
saveURI="s3://billsdata.net/CommonCrawl/hyperlinkgraph/cc-main-2017-may-jun-jul/domaing
#pld_df.coalesce(64).write.save(saveURI) # Use all default options
pld_df=spark.read.load(saveURI)
pld_df.show(3)
pld_df.cache()
#print(pld_df.count()) # Should have 91M domains
```

```
+---+-------+
| ID|    PLD|
+---+-------+
|  0|  aaa.a|
|  1| aaa.aa|
|  2|aaa.aaa|
+---+-------+
only showing top 3 rows
DataFrame[ID: string, PLD: string]
```

```
%pyspark                                                                        FINISHED

# From Paul 5.

# Next, we'll construct a local dictionary from of all the PLDS (key is the PLD, value 
# This is our truth-table of known PLDs that we'll use when counting hosts
# Create a bloom filter using a pure python package (might be a little slow)
from pybloom import BloomFilter
pld_bf = BloomFilter(capacity=91000000, error_rate=0.005)

for row in pld_df.rdd.collect(): # limit(10000000) # TODO: Still bad (and exceeds spark
    pld_bf.add(row['PLD'])

print(pld_df.rdd.take(3))
print(pld_df.rdd.take(3)[2]['PLD'])
#pld_bf.add(pld_df.rdd.take(3)[2]['PLD'])
print("aaa.aaa" in pld_bf) # Should be True

import sys
print(sys.getsizeof(pld_bf))
print(len(pld_bf)) # Should match number of items entered

# Broadcast the bloom filter so it's available on all the slave nodes - we don't need t
# it any more so it's fine being immutable.
pld_bf_distrib=sc.broadcast(pld_bf)

print("aaa.aaa" in pld_bf) # Should be true
print("aaa.aaa.bla" in pld_bf) # Should be false
print("aaa.aaa" in pld_bf_distrib.value) # Should be true
print("aaa.aaa.bla" in pld_bf_distrib.value) # Should be false
```

```
[Row(ID=u'0', PLD=u'aaa.a'), Row(ID=u'1', PLD=u'aaa.aa'), Row(ID=u'2', PLD=u'aaa.aaa')]
aaa.aaa
True
64
90751305
True
False
True
False
```

```
%pyspark                                                                    FINISHED

# From Paul 5.

# Returns a Boolean to say whether PLD is a hostname in itself
def is_a_pld(hostname):
    #if hostname in pld_lookup_table:
    #if pld_lookup_table.filter(lambda a: a == hostname).count()>0:
    if hostname in pld_bf_distrib.value:
        return True
    else:
        return False

# Define a function to do the hostname->pld conversion, if the pld exists in our dictio
def convert_hostname(hostname):
    # Return hostname as-is, if this is already a PLD
    #if hostname in pld_lookup_table:
    #if pld_lookup_table.filter(lambda a: a == hostname).count()>0:
    if hostname in pld_bf_distrib.value:
        return hostname
    # Otherwise we're going to have to split it up and test the parts
    try:
        parts=hostname.split('.')
        if (len(parts)>4 and is_a_pld('.'.join(parts[0:4]))):
            return '.'.join(parts[0:4])
        if (len(parts)>3 and is_a_pld('.'.join(parts[0:3]))):
            return '.'.join(parts[0:3])
        if (len(parts)>2 and is_a_pld('.'.join(parts[0:2]))):
            return '.'.join(parts[0:2])
        if (len(parts)>1 and is_a_pld('.'.join(parts[0:1]))):
            return '.'.join(parts[0:1])
        return "ERROR" # Couldn't find a corresponding PLD - this should never happen!
    except:
        return "ERROR"

udf_convert_hostname = udf(convert_hostname, StringType())

# Test
print(convert_hostname("aaa.aaa"))
print(is_a_pld("aaa.aaa")) # Should be true
```

```
aaa.aaa
True
```

```
%pyspark
```

FINISHED

```
# Function to reverse hostnames
from pyspark.sql.functions import udf
def reverse_domain(domain):
    return '.'.join(reversed(domain.split('.')))
print(reverse_domain("com.facebook"))
udf_reverse_domain = udf(reverse_domain, StringType())

# Convert hosts in Topic DF to PLDs using convert_hostname function from Paul 5.
topic_df2=topic_df.withColumn("pld",udf_reverse_domain(udf_convert_hostname(udf_reverse
topic_df2.show(10)
```

```
facebook.com
+--------------------+--------------------+--------------------+-------------------+----
----------------+-------------------+--------------------+-------------------+--------
------------+
|                host|                 url|              topic1|             score1|
topic2|              score2|              topic3|             score3|                p
ld|
+--------------------+--------------------+--------------------+-------------------+----
----------------+-------------------+--------------------+-------------------+--------
------------+
|  astro.uchicago.edu|http://astro.uchi...|news_article_new_...|0.31914721112987016|said
_like_one_lov...| 0.27035210388505515|university_resear...| 0.14147791332687423|
uchicago.edu|
|blog.greenmountai...|http://blog.green...|said_like_one_lov...| 0.2992614679052542|hote
ls_hotel_reso...| 0.21444496753319794|views_january_day...| 0.17792143718874276|greenmou
ntaininn.com|
|        cleotube.com|http://cleotube.c...|porn_sex_tube_vid...| 0.9935485634990251|isla
nds_republic...|0.003696503953861141|page_wiki_add_com...|2.863101549746408   |
```

```
%pyspark
```

FINISHED

```
# If multiple rows per PLD, only use the one with the shortest URL.
def url_len(url):
    return len(url)
print(url_len("com.facebook.bla"))
udf_url_len = udf(url_len, IntegerType())
from pyspark.sql.functions import col, desc
topic_df3=topic_df2.withColumn("url_len", udf_url_len("url"))
#topic_df3.show(30)

# Use window functions to filter to only the row with the shortest URL for each PLD
from pyspark.sql.window import Window
from pyspark.sql.functions import rank, col
window = Window.partitionBy(topic_df3['pld']).orderBy(topic_df3['url_len']) # Ascending
topic_df4=topic_df3.select('*', rank().over(window).alias('rank')).filter(col('rank') <
topic_df4.show(20)

# TODO: Consider putting the three topic columns into a single column.
```

```
load_free_sea...|  0.1264106404199821|david_john_michae...| 0.04685124133770203|       b
ehairy.com|     64|   1|
|         behairy.com|http://behairy.co...|porn_sex_tube_vid...| 0.8249370983378203|down
load_free_sea...|  0.1180124766479175|david_john_michae...| 0.05537182630503831|       b
ehairy.com|     64|   1|
|www.chester-jense...|http://www.cheste...|david_john_michae...| 0.2645494348182479|inc_
llc_products_...|  0.2603211953958203|document_geteleme...| 0.23348354412455652|chester-
jensen.com|     40|   1|
|         www.cmu.edu|http://www.cmu.ed...|university_resear...| 0.7930618164877749|new_
john_lhblk_st...| 0.10985124039316259|views_january_day...| 0.05310735450633527|
cmu.edu|     33|   1|
|         www.cmu.edu|http://www.cmu.ed...|university_resear...| 0.7919300397654346|cell
_protein_sex_...| 0.15335284335516533|david_john_michae...|  0.0262016766978846|
cmu.edu|     33|   1|
|   www.com2media.com|http://www.com2me...|information_servi...|0.38418043058257784|com_
game_games_se...| 0.19748302138113874|use_site_informat...| 0.19251118010797869|     com
2media.com|     41|   1|
|    www.cuballot.com|https://www.cubal...|information_servi...|   0.78118276597855|univ
```

```
%pyspark
```

FINISHED

```
# Join on host/PLD
pld_df.unpersist()
topic_df2.unpersist()
topic_df3.unpersist()
enrich_summary_df=example_summary_df.join(topic_df4, example_summary_df.payLevelDomain=
                                    .drop("host").drop("url").drop("score1").drop("scor
                                    .withColumnRenamed("topic1","exampleIndexPageTopics
enrich_summary_df.sort("numHosts",ascending=False).show()
enrich_summary_df.count()
```

```
------+
|      uchastings.edu|      99|          true|      1986|       2131|          true|99.94
3|99.946|uchastings.edu, a...|anu.edu.au, backe...|coronertalk.com, ...|  score_com_link
_en...|
|            semo.edu|      96|          true|      1923|       2636|          true|99.98
7|99.939|semo.edu, 6.semo....|blogspot.com, cap...|dineoncampus.com,...|  university_res
ear...|
|             rpi.edu|     945|          true|     15109|       9235|          true|99.99
3|99.993|rpi.edu, 3helix.r...|boinc.cat, succes...|partsavatar.ca, s...|  university_res
ear...|
|           varvar.ru|       9|          true|       317|        691|          true|98.76
2|97.972|varvar.ru, univer...|lacamorra.ru, sig...|foto-planeta.com,...|  new_john_lhblk
_st...|
|   jhinvestments.com|       9|          true|       144|         37|          true|92.37
7|99.804|jhinvestments.com...|blogspot.com, bos...|iinews.com, leadf...|  score_com_link
_en...|
|           varvar.ru|       9|          true|        58|        283|          true|98.76
2|97.972|varvar.ru, univer...|lacamorra.ru, sig...|foto-planeta.com,...|  new_john_lhblk
```

```
%pyspark

# Save Example Summaries as GZIP files, approx 100MB each.
outURI="s3://billsdata.net/CommonCrawl/domain_summaries_withtopics2/"
codec="org.apache.hadoop.io.compress.GzipCodec"
enrich_summary_df.coalesce(1).write.format('com.databricks.spark.csv').options(header='
```

FINISHED

```
%pyspark
```

FINISHED