Tom 1 Topic Mode...

FINISHED

```
%pyspark

# PySpark CommonCrawl Topic Modelling
# Tom V - 21/10/2017

import boto
from boto.s3.key import Key
from gzipstream import GzipStreamFile
from pyspark.sql.types import *
import warc
import ujson as json
from urlparse import urlparse
from langdetect import detect_langs

wetlist = sc.textFile("s3://commoncrawl/crawl-data/CC-MAIN-2017-04/wet.paths.gz")
wetlist.cache()

def unpack(uri):
    conn = boto.connect_s3(anon=True, host='s3.amazonaws.com')
    bucket = conn.get_bucket('commoncrawl')
    key_ = Key(bucket, uri)
    file_ = warc.WARCFile(fileobj=GzipStreamFile(key_))
    return file_

def detect(x):
    try:
        return detect_langs(x[:300])[0].lang
    except Exception as e:
        return None

def process_wet(id_, iterator):
    for uri in iterator:
        file = unpack(uri)
        for record in file:
            try:
                #url = record.rec_headers.get_header('WARC-Target-URI')
                #yield record, record.content_stream().read().decode('utf-8')
                url = record.url
                domain = None if not url  else urlparse(url).netloc
                text = record.payload.read().decode('utf-8')
                lang = detect(text)
                yield domain, url, lang, text
            except Exception as e:
                yield e
```

RUNNING 0%

```
%pyspark
# PARAMETER - number of input files
nfiles = 256
```

```
# PARAMETER - slices / partitions of input
files = sc.parallelize(wetlist.take(nfiles), numSlices=64)

# TODO: Make this use more than one CPU!
print(files.getNumPartitions())
rdd = files.mapPartitionsWithIndex(process_wet)

docs = rdd.toDF(["domain", "url", "lang", "text"])
docs.cache()

docs.show()
```

```
64
```

Started a minute ago.

```
%pyspark

docs_en = docs.filter(docs.lang == 'en')

# PARAMETER - possibly set partitions?
docs_en = docs_en.repartition(64)
```

PENDING

```
%pyspark

stopwords_english = ['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves',
    'you', 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself',
    'she', 'her', 'hers', 'herself', 'it', 'its', 'itself', 'they', 'them', 'their',
    'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'these',
    'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has',
    'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but',
    'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with',
    'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after',
    'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over',
    'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where',
    'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some'
    , 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very',
    's', 't', 'can', 'will', 'just', 'don', 'should', 'now', 'd', 'll', 'm', 'o', 're'
    , 've', 'y', 'ain', 'aren', 'couldn', 'didn', 'doesn', 'hadn', 'hasn', 'haven',
    'isn', 'ma', 'mightn', 'mustn', 'needn', 'shan', 'shouldn', 'wasn', 'weren', 'won'
    , 'wouldn']

from pyspark.ml import Pipeline
from pyspark.ml.feature import RegexTokenizer, CountVectorizer, StopWordsRemover

# PARAMETER - regex tokenization
tokenizer = RegexTokenizer(inputCol="text", outputCol="words", pattern="\\p{L}{3,}",
```

PENDING

```
+--------------------+--------------------+----+--------------------+--------------------
-+--------------------+--------------------+
|              domain|                 url|lang|                text|                word
s|            filtered|                 vec|
+--------------------+--------------------+----+--------------------+--------------------
-+--------------------+--------------------+
|            1337x.to|http://1337x.to/t...|  en|Download Bhindi B...|[download, bhindi..
.|[download, bhindi...|(20000,[0,1,3,5,6...|
|       909sickle.net|http://909sickle....|  en|Double Oh Duty
ab...|[double, duty, ab...|[double, duty, sa...|(20000,[5,16,53,7...|
|          adioso.com|http://adioso.com...|  en|The best flights ...|[the, best, fligh..
.|[best, flights, f...|(20000,[0,5,6,10,...|
|   akihabaranews.com|http://akihabaran...|  en|| AkihabaraNews S...|[akihabaranews, s..
.|[akihabaranews, s...|(20000,[0,5,7,9,1...|
| alternativesite.net|http://alternativ...|  en|Alternative for 1...|[alternative, for..
.|[alternative, pre...|(20000,[0,1,3,5,1...|
| animaldiversity.org|http://animaldive...|  en|ADW: Apotomops: C...|[adw, apotomops, ..
.|[adw, apotomops, ...|(20000,[5,17,19,2...|
```

```
%pyspark                                                                    PENDING

#Run the topic modelling

from pyspark.ml.clustering import LDA
#inputCol="vec", outputCol="ldaVec", k=3, optimizer="online"

# Fix java memory errors, perhaps using:
# spark.driver.memory 256g - DIDN'T WORK
# or by reducing vocabSize from 100k to 20k - WORKS!

lda = LDA(k=100, maxIter=100, featuresCol="vec")
ldaModel = lda.fit(vecs)
print(ldaModel.isDistributed())
```

```
Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-5732463703618195166.py", line 367, in <module>
    raise Exception(traceback.format_exc())
Exception: Traceback (most recent call last):
  File "/tmp/zeppelin_pyspark-5732463703618195166.py", line 360, in <module>
    exec(code, _zcUserQueryNameSpace)
  File "<stdin>", line 4, in <module>
AttributeError: 'LDA' object has no attribute 'isDistributed'
```

```
%pyspark                                                                    PENDING

# Save the models
ldaModel.save('s3://billsdata.net/CommonCrawl/topic_model_256files/ldamodel')
pipeline.save('s3://billsdata.net/CommonCrawl/topic_model_256files/textpipeline')
```

```
%pyspark
```

PENDING

```
# Get topic vectors for index pages (estimate of topic vec per domain)

vecs_index = vecs.filter("url LIKE '%index.html'")
results = ldaModel.transform(vecs_index)

# Drop text cols
results2=results.drop('text').drop('words').drop('filtered')

# Save domain topic vecs
results2.write.parquet('s3://billsdata.net/CommonCrawl/topic_model_256files
```

PENDING

```
%pyspark

# Create a dataset containing just the host, url and top 3 topic labels & scores

import pandas as pd
import numpy as np
topicIndices = ldaModel.describeTopics(maxTermsPerTopic = 5).collect()
vocab = model.stages[2].vocabulary

topic_labels = []
for i, (topic, terms, termWeights) in enumerate(topicIndices):
    topwords = pd.Series(dict(zip([vocab[t] for t in terms], termWeights
        ))).sort_values(ascending=False)
    topic_labels.append('_'.join(topwords.index.values))

topic_labels = np.array(topic_labels)

def topTopics(x):
    labels = topic_labels[np.argsort(x.topicDistribution)[::-1][:3]]
    scores = np.sort(x.topicDistribution)[::-1][:3]
    return (x.domain, x.url, str(labels[0]), float(scores[0]), str(labels[1]), float
        (scores[1]), str(labels[2]), float(scores[2]))

results3 = results2.rdd.map(topTopics)
results3 = results3.toDF(["host", "url", "topic1", "score1", "topic2", "score2",
    "topic3", "score3"])
```

```
+--------------------+--------------------+--------------------+--------------------+----
----------------+--------------------+--------------------+--------------------+
|                host|                 url|              topic1|              score1|
topic2|              score2|              topic3|              score3|
+--------------------+--------------------+--------------------+--------------------+----
----------------+--------------------+--------------------+--------------------+
|        nowretro.com|http://nowretro.c...|porn_sex_tube_big...| 0.9989964613360407|emai
l_search_name...|1.934827854166181...|like_one_would_ge...|1.520965688788838E-5|
| onlyamateurtube.com|http://onlyamateu...|porn_sex_tube_big...| 0.5909796072765714|view
s_videos_vide...| 0.32828038918474856|food_recipes_reci...| 0.04096231431567997|
|       money.cnn.com|http://money.cnn....|market_energy_com...| 0.3001351025451066|said
_people_news_...| 0.16049798144775618|offer_credit_bank...|  0.1006166940895161|
|       idlebrain.com|http://idlebrain....|art_movie_movies_...| 0.6849752482814744|phot
o_image_herit...| 0.13877600868871826|com_http_www_org_net|  0.1367208427098497|
|      lvx.tripod.com|http://lvx.tripod...|church_god_bible_...| 0.6182698205816841|ship
ping_order_pr...|  0.1357727539039878|tutoring_one_also...|   0.127051518655225|
|lesbian-sapphic-e...|http://lesbian-sa...|porn_sex_tube_big...| 0.9975070296105998|chur
ch_god_bible_...|9.614505413466535E-4| dec_nov_oct_mar_feb|7.581462548444677E-4|
```

```
%pyspark
```

FINISHED