#!/usr/bin/env python

# um

from \_\_future\_\_ import print\_function

import re

import sys

import json

import fileinput

import os

# David: Ohne das hat es bei mir nicht funktioniert

import ssl

ssl.\_create\_default\_https\_context = ssl.\_create\_unverified\_context

# # zum ausschliessen von worten mit umlauten.

# os.environ["PYTHONIOENCODING"] = "utf-8";

def main():

try:

from urllib import urlopen # Python 2

except ImportError:

from urllib.request import urlopen # Python 3

MAX\_WORDS = 60

word\_counts = {}

stop\_words = set([

"wuppen", "wuppt", "wuppst", "wuppte", "und", "die", "das", "hat", "oder", "ist",

"mich", "auch", "einen", "dir", "machen", "euch", "sind", "noch", "nicht", "dann", "sollen",

"aber", "weil", "mehr", "diese", "doch", "wird", "den", "wie", "sich", "eine",

"nichts", "der", "drei", "dass", "auf", "uns", "haben", "von", "mal", "müssen",

"bist", "finde", "meine", "geht", "für", "dafür", "mit", "dieser", "wenn", "darf",

"aus", "tun", "damit", "war", "hier", "etwas", "denn", "hätten", "bin", "dem", "ihre", "mir",

"sein", "kann", "wäre", "zum", "mein", "seit", "werden", "wieder", "macht", "ziemlich",

"nur", "halt", "bis", "davon", "ihnen", "was", "gar", "ihr", "also", "eigentlich", "bekommen",

"als", "denen", "kommen", "unserem", "hab", "ohne", "soll", "echt", "habe", "zur", "eben", "kriegen",

"dich", "oft", "hin", "nun", "sagen", "vor", "ganzen", "natürlich", "ihren", "wohl", "werde",

"sonst", "klar", "weg", "ein", "viel", "viele", "bei", "einer", "können", "sie", "werd", "meiner",

"wurde", "schon", "her", "rum", "gibt", "selber", "dein", "will", "raus", "sondern",

"alle", "nix", "keine", "selbst", "gerade", "drauf", "steht", "unsere", "gab", "nächste", "waren",

"kaum", "muss", "wann", "wer", "genug", "einem", "ass", "trotzdem", "darauf", "nach", "gehts",

"gewuppt", "könnte", "braucht", "geben", "dran", "hast", "statt", "plus", "fest", "ach", "kein"

"dieses", "fast", "bekommt", "sehen", "dieses", "erst", "ich", "wir"

])

for line in fileinput.input():

try:

tweet = json.loads(line)

except:

pass

for word in text(tweet).split(' '):

word = word.lower()

word = word.replace(".", "")

word = word.replace(",", "")

word = word.replace("...", "")

word = word.replace("'", "")

word = word.replace(":", "")

word = word.replace("(", "")

word = word.replace(")", "")

word = word.replace("!", "")

word = word.replace("?", "")

#print(word)

if len(word) < 3: continue

if len(word) > 15: continue

if word in stop\_words: continue

if word[0] in ["@", "#"]: continue

if re.match('https?', word): continue

if word.startswith("rt"): continue

if not re.match('^[a-z]', word, re.IGNORECASE): continue

word\_counts[word] = word\_counts.get(word, 0) + 1

sorted\_words = list(word\_counts.keys())

sorted\_words.sort(key = lambda x: word\_counts[x], reverse=True)

top\_words = sorted\_words[0:MAX\_WORDS]

words = []

count\_range = word\_counts[top\_words[0]] - word\_counts[top\_words[-1]] + 1

size\_ratio = 100.0 / count\_range

for word in top\_words:

size = int(word\_counts[word] \* size\_ratio) + 15

words.append({

"text": word,

"size": size

})

wordcloud\_js = urlopen('https://raw.githubusercontent.com/jasondavies/d3-cloud/master/build/d3.layout.cloud.js').read()

output = """<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>twarc wordcloud</title>

<script src="https://d3js.org/d3.v3.min.js"></script>

</head>

<body>

<script>

// embed Jason Davies' d3-cloud since it's not available in a CDN

%s

var fill = d3.scale.category20();

var words = %s

d3.layout.cloud().size([800, 800])

.words(words)

.rotate(function() { return ~~(Math.random() \* 2) \* 90; })

.font("Impact")

.fontSize(function(d) { return d.size; })

.on("end", draw)

.start();

function draw(words) {

d3.select("body").append("svg")

.attr("width", 1000)

.attr("height", 1000)

.append("g")

.attr("transform", "translate(400,400)")

.selectAll("text")

.data(words)

.enter().append("text")

.style("font-size", function(d) { return d.size + "px"; })

.style("font-family", "Impact")

.style("fill", function(d, i) { return fill(i); })

.attr("text-anchor", "middle")

.attr("transform", function(d) {

return "translate(" + [d.x, d.y] + ")rotate(" + d.rotate + ")";

})

.text(function(d) { return d.text; });

}

</script>

</body>

</html>

""" % (wordcloud\_js.decode('utf8'), json.dumps(words, indent=2))

sys.stdout.write(output)

def text(t):

if 'full\_text' in t:

return t['full\_text']

return t['text']

if \_\_name\_\_ == "\_\_main\_\_":

main()