```python
# STRUCTURED DATA PROCESSING

import csv
import numpy as np
import pandas as pd

infile = 'ted_talks_en.csv'

# create new empty list
IDlist = []

with open(infile, 'r') as csvfile:
    # the csv file reader returns a list of the csv items on each line
    IDreader = csv.reader(csvfile, dialect='excel', delimiter=',')

    # from each line, a list of row items, put each element in a diction
ary with a key representing the data
    for line in IDreader:
      #skip lines without data, specific for each file to catch non-data
lines
        if line[0] == '' or line[0].startswith('talk_id'):
            continue
        else:
            try:
              # create a dictionary for each ID
              ID = {}
              # add each piece of data under a key representing that data
              ID['Talk ID'] = line[0]
              ID['Title'] = line[1]
              ID['Speaker 1'] = line[2]
              ID['All Speakers'] = line[3]
              ID['Occupations'] = line[4]
              ID['About Speakers'] = line[5]
              ID['Views'] = line[6]
              ID['Date Recorded'] = line[7]
              ID['Date Published'] = line[8]
              ID['Event'] = line[9]
              ID['Lang Native'] = line[10]
              ID['Lang Available'] = line[11]
              ID['Comments'] = line[12]
              ID['Duration'] = line[13]
              ID['Topics'] = line[14]
              ID['Related Talks'] = line[15]
              ID['URL'] = line[16]
              ID['Description'] = line[17]
              ID['Transcript'] = line[18]

              # add this ID to the list
              IDlist.append(ID)
            # catch errors in file formatting (number items per line)  and
print an error message
            except IndexError:
              print ('Error: ', line)
csvfile.close()

df = pd.DataFrame(IDlist, columns = ['Talk ID', 'Title', 'Speaker 1', 'O
```

```python
ccupations', 'About Speakers', 'Views',
                                    'Date Recorded', 'Date Published',
'Comments', 'Duration', 'Topics',
                                    'Related Talks', 'Description', 'Tr
anscript'])

# Convert columns to appropriate data types

# strip blank spaces from 'Talk ID' column
df['Talk ID'] = df['Talk ID'].str.strip()
# add leading zeros to 'Talk ID' column
df['Talk ID'] = df['Talk ID'].str.rjust(4, "0")
# set df index to 'Talk ID' field
df = df.set_index('Talk ID').sort_values(by = 'Talk ID')

df['Views'] = df['Views'].astype(int)

# convert dates to datetime format using pd.to_datetime()
df['Date Recorded'] = pd.to_datetime(df['Date Recorded'], format = '%Y-%
m-%d')
df['Date Published'] = pd.to_datetime(df['Date Published'], format = '%Y
-%m-%d')

# define replace function
def item_replace(xstr):
    return xstr.replace('','0') # in a string, replace any occurrence of
 '' with '0'

# replace blank with 0 in 'Comments' column
df['Comments'] = df['Comments'].map(item_replace).astype(int)

# convert seconds to hours:minutes:seconds using pd.to_datetime()
df['Duration'] = df['Duration'].astype(int)
df['Duration'] = pd.to_datetime(df['Duration'], unit='s').dt.strftime("%
H:%M:%S")
df
```

Out[1]:

Talk ID	Title	Speaker 1	Occupations	About Speakers	Views	Date Recorded	Date Published	Cor
0001	Averting the climate crisis	Al Gore	{0: ['climate advocate']}	{0: 'Nobel Laureate Al Gore focused the world'...	3523392	2006-02-25	2006-06-27	
0002	Simple designs to save a life	Amy Smith	{0: ['inventor', 'engineer']}	{0: 'Amy Smith designs cheap, practical fixes ...	1724438	2006-02-24	2006-08-15	
0003	How to rebuild a broken state	Ashraf Ghani	{0: ['president-elect of afghanistan']}	{0: 'Ashraf Ghani, Afghanistan's new president...	981920	2005-07-12	2006-10-18	
0004	The real future of space exploration	Burt Rutan	{0: ['aircraft engineer']}	{0: "In 2004, legendary spacecraft designer Bu...	2427994	2006-02-24	2006-10-25	
0005	Great cars are great art	Chris Bangle	{0: ['car designer']}	{0: 'Car design is a ubiquitous but often over...	978483	2002-02-02	2007-04-05	
...	...	...	...	...	...	...	...	...
9985	"My Fine Reward"	Tito Deler	{0: ['blues musician', 'graphic designer']}	{0: 'Tito Deler is a soulful musician with a s...	221834	2017-11-14	2018-02-09	
9986	This company pays kids to do their math homework	Mohamad Jebara	{0: ['education entrepreneur']}	{0: 'Mohamad Jebara is the founder and CEO of ...	1128352	2017-12-11	2018-02-08	
9987	How to fix a broken heart	Guy Winch	{0: ['psychologist', 'author']}	{0: 'Guy Winch asks us to take our emotional h...	10235040	2017-04-24	2018-02-05	
9988	The surprising ingredient that makes businesse...	Marco Alverà	{0: ['businessman', 'fairness crusader']}	{0: 'Marco Alverà is an Italian/American busin...	2855104	2017-10-04	2018-02-13	

Talk ID	Title	Speaker 1	Occupations	About Speakers	Views	Date Recorded	Date Published	Cor
9989	Fashion that celebrates African strength and s...	Walé Oyéjidé	{0: ['designer', 'writer', 'musician', 'lawyer']}	{0: 'TED Fellow Walé Oyéjidé combats bias with...	904845	2017-08-27	2018-02-15	

4005 rows × 13 columns

```
In [2]: # Are there any speakers than gave more than one talk?
        speakers_max = df[['Title']].groupby(df['Speaker 1']).count()
        speakers_max.sort_values(by = 'Title', ascending = False)
```

Out[2]:

Speaker 1	Title
Alex Gendler	34
Iseult Gillespie	19
Emma Bryce	12
Daniel Finkel	11
Hans Rosling	10
...	...
Giada Gerboni	1
Gian Giudice	1
Gil Weinberg	1
Giles Duley	1
Özlem Cekic	1

3274 rows × 1 columns

In [3]:
```python
# what is the most common recording date? (month, year)
recording_max = df[['Title', 'Date Recorded']].groupby(['Date Recorded'
]).count()
recording_max.sort_values(by = 'Title', ascending = False)
```

Out[3]:

| | Title |
Date Recorded	
2017-04-24	97
2018-04-10	95
2019-04-15	90
2007-03-03	54
2017-08-27	53
...	...
2014-03-16	1
2014-03-24	1
2014-03-25	1
2014-03-27	1
2020-04-30	1

1334 rows × 1 columns

```
In [4]:  # is there a pattern with publishing date?
         publishing_max = df[['Title', 'Date Published']].groupby(['Date Published']).count()
         publishing_max.sort_values(by = 'Title', ascending = False)
```

Out[4]:

	Title
Date Published	
2017-09-07	27
2019-02-15	24
2019-02-12	23
2019-03-15	22
2019-04-01	22
...	...
2013-04-25	1
2013-04-26	1
2013-04-29	1
2013-04-30	1
2014-01-24	1

2962 rows × 1 columns

```
In [5]:  # which talk has the most number of views?
         df[['Views']].mean() # average number of views is 2.148006e+06 -> 2,148,006
         max_views = df.sort_values(by = 'Views', ascending = False)[:1]
         max_views
```

Out[5]:

	Title	Speaker 1	Occupations	About Speakers	Views	Date Recorded	Date Published	Comments
Talk ID								
0066	Do schools kill creativity?	Sir Ken Robinson	{0: ['author', 'educator']}	{0: "Creativity expert Sir Ken Robinson challe...	65051954	2006-02-25	2006-06-27	40903010

```
In [6]:  # which talk has the most number of comments? least comments? avg commen
         ts?
         df[['Comments']].mean() # average number of comments is 239073.907615
         max_comments = df.sort_values(by = 'Comments', ascending = False)[:1]
         max_comments
```

Out[6]:

Talk ID	Title	Speaker 1	Occupations	About Speakers	Views	Date Recorded	Date Published	Comments	Dura
0113	Militant atheism	Richard Dawkins	{0: ['evolutionary biologist']}	{0: 'Oxford professor Richard Dawkins has help...	5788514	2002-02-02	2007-04-16	60404090	00:2

```
In [7]:  # which talk has the longest duration?
         df.sort_values(by = 'Duration', ascending = False)[:1]
```

Out[7]:

Talk ID	Title	Speaker 1	Occupations	About Speakers	Views	Date Recorded	Date Published	Comments	Du
54715	How we can turn the tide on climate	Chris Anderson	{0: ['head of ted'], 1: ['climate advocate']}	{0: 'After a long career in journalism and pub...	1493370	2019-12-04	2019-12-12	4040	01

```
In [8]:   # transcript analysis; word cloud, sentiment analysis
          ted_transcript = df[['Title', 'Speaker 1', 'Transcript']]
          ted_transcript
```

Out[8]:

Talk ID	Title	Speaker 1	Transcript
0001	Averting the climate crisis	Al Gore	Thank you so much, Chris. And it's truly a gre...
0002	Simple designs to save a life	Amy Smith	In terms of invention, I'd like to tell you th...
0003	How to rebuild a broken state	Ashraf Ghani	A public, Dewey long ago observed, is constitu...
0004	The real future of space exploration	Burt Rutan	I want to start off by saying, Houston, we hav...
0005	Great cars are great art	Chris Bangle	What I want to talk about is, as background, i...
...	...	...	...
9985	"My Fine Reward"	Tito Deler	(Music) Sun shining up above down here it's 10...
9986	This company pays kids to do their math homework	Mohamad Jebara	For as long as I remember, I've loved mathemat...
9987	How to fix a broken heart	Guy Winch	At some point in our lives, almost every one o...
9988	The surprising ingredient that makes businesse...	Marco Alverà	For me, it was not being invited to a friend's...
9989	Fashion that celebrates African strength and s...	Walé Oyéjidé	It is often said that the stories of history a...

4005 rows × 3 columns

```
In [9]:   # SEMI-STRUCTURED DATA PROCESSING

          # This program reads in JSON formatted data from a MongoDB collection.
          # This is in a format that is structured with lines of data representing
          one Tweet for Twitter.
          # This program contains the data as lists of JSON structures, which are
           just Python dictionaries and lists.

          # START MONGODB
          # brew services start mongodb-community@4.4

          # SCRAPE TWEETS 2020 0829
          # !python run_twitter_simple_search_save.py "#TED" 4000 ted tedtweets
          # !python run_twitter_simple_search_save.py "#tedtalk" 4000 ted tedtweet
          s
          # !python run_twitter_simple_search_save.py "#TEDx" 4000 ted tedtweets

          # reran and collected more tweets in anticipation of only analyzing twee
          t text with 'lang'== 'en'
```

```
In [10]: import pymongo
         client = pymongo.MongoClient('localhost', 27017)

         db = client.ted

         db.list_collection_names()
```

Out[10]: ['tedtweets']

```
In [11]: collection = db.tedtweets

         tweets = collection.find()

         tweetlist = [tweet for tweet in tweets]
         len(tweetlist) # as of 08/29/2020
```

Out[11]: 4601

```
In [12]:   # Here is a little print function that will help.

           def print_tweet_data(tweets):
               for tweet in tweets:
                   print('\nDate:', tweet['created_at'])
                   print('From:', tweet['user']['name'])
                   print('Message:', tweet['text'])
                   if not tweet['place'] is None:
                       print('Place:', tweet['place']['full_name'])

           print_tweet_data(tweetlist[:5])
```

```
Date: Sat Aug 29 19:07:13 +0000 2020
From: ✨ Petite Quarterback ✨
Message: My college students are making their own Bitmoji classrooms an
d I absolutely love it! #BMCC #TED
Place: Manhattan, NY

Date: Sat Aug 29 18:48:48 +0000 2020
From: ~Bella~
Message: PARTY ON DUDES.                          #BillAndTed #bill #ted
#billandtedfancam #fancam #fancams #billandtedfancams https://t.co/5MyC
whhD0D

Date: Sat Aug 29 18:35:04 +0000 2020
From: polish-hive
Message: Kolonizacja Marsa a praca w sektorze kosmicznym - moje wystąpi
enie na TEDx

#hive #posh #Poland #Polska #space… https://t.co/e1KbJDCfGK

Date: Sat Aug 29 17:57:10 +0000 2020
From: TED DiyarbakırKoleji
Message: 30 Ağustos Zafer Bayramı kutlu olsun.
#TED #Diyarbakır #30ağustos https://t.co/hx2yNO9sUc

Date: Sat Aug 29 17:47:50 +0000 2020
From: Présentation Pertinente
Message: Concevoir un support de présentation peut-être comparé à une o
ssature, comme évoqué dans le guide officiel #TED - P… https://t.co/tnk
q63hVG8
```

```
In [13]:  # My program contains pandas dataframes for processed data.

          # This program does some processing to collect data from some of the fie
          lds the questions described below,
          # and write a file with the data suitable for answering each question.

          import numpy as np
          import pandas as pd

          df = pd.DataFrame(tweetlist)

          # Test for null values and remove optional fields
          df.isna().sum() # sum of NaN
```

```
Out[13]:  _id                         0
          created_at                  0
          id                          0
          id_str                      0
          text                        0
          truncated                   0
          entities                    0
          metadata                    0
          source                      0
          in_reply_to_status_id       4371
          in_reply_to_status_id_str   4371
          in_reply_to_user_id         4338
          in_reply_to_user_id_str     4338
          in_reply_to_screen_name     4338
          user                        0
          geo                         4593
          coordinates                 4593
          place                       4514
          contributors                4601
          is_quote_status             0
          retweet_count               0
          favorite_count              0
          favorited                   0
          retweeted                   0
          lang                        0
          extended_entities           4380
          possibly_sensitive          1788
          quoted_status_id            4101
          quoted_status_id_str        4101
          quoted_status               4422
          retweeted_status            2143
          dtype: int64
```

```
In [14]: df.dropna()

         # select columns
         df = df[['_id',
                  'created_at',
                  'text',
                  'entities', ### hashtag from entities
                  'user', ### name from user
                  'retweet_count',
                  'favorite_count',
                  'lang']]

         # set index to _id column
         # df = df.set_index('_id')
```

```
In [15]: # convert df[['created_at']] to string;
         df[['created_at']] = df[['created_at']].astype(str)

         # convert df[['created_at']] to datetime using pd.to_datetime()
         df[['created_at']] = pd.to_datetime(df['created_at'], format = '%a %b %d
         %H:%M:%S +0000 %Y')
```

```
In [16]: # convert counts from str to int type
         df[['retweet_count']] = df[['retweet_count']].astype(int)
         df[['favorite_count']] = df[['favorite_count']].astype(int)
```

```
In [17]: # bin Tweets by day
         created_date = df['created_at'].dt.date
         df['created_date'] = created_date

         # bin Tweets by hour
         created_hour = df['created_at'].dt.hour
         df['created_hour'] = created_hour
```

```
In [18]:  # report on the number of Tweets per day
          df[['created_date', 'created_at']].groupby(['created_date']).count()
```

Out[18]:

	created_at
created_date	
2020-08-21	426
2020-08-22	784
2020-08-23	588
2020-08-24	492
2020-08-25	507
2020-08-26	604
2020-08-27	425
2020-08-28	442
2020-08-29	333

```python
# What is being Tweeted on 08-22-2020?
df['created_date'] = df['created_date'].astype(str)
max_date = df[(df['created_date'] > '2020-08-21') & (df['created_date']
< '2020-08-23')]
max_date[['created_date', 'text']][:50]

# Do Schools Kill Creativity? - Sir Ken Robinson (1950-03-04 - 2020-08-2
1)
# https://www.ted.com/talks/sir_ken_robinson_do_schools_kill_creativity?
language=en
```

Out[19]:

	created_date	text
1021	2020-08-22	RT @TEDxCDMX: Hoy más que nunca, recordemos es...
1022	2020-08-22	그럼 어째서 우리는 결코 사무엘 피어퐁 랭리에 대해서 들어본 적이 없는가? \nTh...
1023	2020-08-22	RT @fahadalahmdi: هل تعرف ماهو الفرق بين السؤا...
1024	2020-08-22	@SirKenRobinson murió ayer 21 agosto 2020, gra...
1025	2020-08-22	RT @thewetbaguett3: Like for part 7! #ted #mov...
1026	2020-08-22	그들은 없었다. 우리가 성공의 레서피라 여기는 것들을 \nthey had none ...
1027	2020-08-22	Like for part 7! #ted #movie #moviescene #4u #...
1028	2020-08-22	ダニエル・H・コーエン: よい議論をするために\nhttps://t.co/llqGnURP...
1029	2020-08-22	RT @TEDxCDMX: Hoy más que nunca, recordemos es...
1030	2020-08-22	If you've never watched @SirKenRobinson's orig...
1031	2020-08-22	@NWSBoulder Who's #ted
1032	2020-08-22	Hoy más que nunca, recordemos esta charla en m...
1033	2020-08-22	RT @TEDxUFPR: Novo design, novo tema, novos pa...
1034	2020-08-22	RT @markycollector: We've had some epic new me...
1035	2020-08-22	@SirKenRobinson has been the most influential ...
1036	2020-08-22	Nos hemos quedado sin @SirKenRobinson He visto...
1037	2020-08-22	Let's honour @SirKenRobinson's life, work and ...
1038	2020-08-22	A truly creative and pioneering professional h...
1039	2020-08-22	Novo design, novo tema, novos palestrantes, NO...
1040	2020-08-22	RT @fahadalahmdi: هل تعرف ماهو الفرق بين السؤا...
1041	2020-08-22	RT https://t.co/nGHtEO8Flm | AKP'den 'Fitch' t...
1042	2020-08-22	RT https://t.co/EMGuud7IVa | AB'den Libya'daki...
1043	2020-08-22	RT https://t.co/PAFPeZ1HM5 | Kocaeli'de fabrik...
1044	2020-08-22	RT https://t.co/hoSM8iptGs | Sivas'ta düğün sa...
1045	2020-08-22	RT https://t.co/ChevIW78nN | Sınır Tanımayan G...
1046	2020-08-22	RT https://t.co/jSahaoE7QE | Akşener: İstanbul...
1047	2020-08-22	RT https://t.co/oEmHxfn5Qn | Meral Akşener: Ga...
1048	2020-08-22	RT https://t.co/pt6XKFbRW7 | Necati Özkan sorm...
1049	2020-08-22	RT https://t.co/CRnpc2rV7Q | İzmir ve Antalya'...
1050	2020-08-22	RT https://t.co/ZNMNzBNlmf | İstanbul'da "Takv...
1051	2020-08-22	RT https://t.co/dOYkvgDsWc | Fiskobirlik, fınd...
1052	2020-08-22	RT https://t.co/ppNrJPxhjc | İBB, deprem odakl...
1053	2020-08-22	RT https://t.co/S7eSYTtqnY | Lukaşenko: Grev k...
1054	2020-08-22	RT https://t.co/dOAMH2fcm3 | Trabzonspor, Flav...

	created_date	text
1055	2020-08-22	RT https://t.co/UNXIY8YzsB | Giresun'da sağana...
1056	2020-08-22	RT https://t.co/QDhYec11Dh | Ölüm orucundaki a...
1057	2020-08-22	RT https://t.co/yQCvWAmhhD | Yavru köpeği ezdi...
1058	2020-08-22	RT https://t.co/T8NrWlMC5r | Şanlıurfa'da mask...
1059	2020-08-22	RT https://t.co/Te4UwwyBGp | Adıgüzel: Sağlık ...
1060	2020-08-22	RT https://t.co/iJP7Rr0Bg2 | Deprem toplantısı...
1061	2020-08-22	RT https://t.co/L2DGqCgEUH | Antalya Serik'te ...
1062	2020-08-22	RT https://t.co/jTA9OCQecP | "Bana buluşmaz de...
1063	2020-08-22	RT https://t.co/Ac12pFtb5z | Neyi güçlendirmel...
1064	2020-08-22	RT @fahadalahmdi: هل تعرف ماهو الفرق بين السؤا...
1065	2020-08-22	His #Ted talks were a great inspiration for me...
1066	2020-08-22	RT @paddyjansen: In memory of one of #educatio...
1067	2020-08-22	In memory of one of #education's greatest insp...
1068	2020-08-22	YouAccel #Media Shared this Video on YouAccel:...
1069	2020-08-22	Following the sad news of the death of Sir Ken...
1070	2020-08-22	Sad news that Sir Ken Robinson has passed away...

```
In [20]:  # report on the number of Tweets per hour
          df[['created_at', 'created_hour']].groupby(['created_hour']).count()
```

Out[20]:

	created_at
created_hour	
0	163
1	68
2	98
3	165
4	123
5	102
6	162
7	287
8	170
9	232
10	146
11	254
12	312
13	259
14	274
15	275
16	214
17	196
18	272
19	163
20	216
21	191
22	133
23	126

```
In [21]:  # report on the number of Tweets per day, per hour
          created_date_hour = df[['created_at', 'created_date', 'created_hour']].g
          roupby(['created_date', 'created_hour']).count()
          created_date_hour

          max_date[['created_at', 'created_date', 'created_hour']].groupby(['creat
          ed_date', 'created_hour']).count().sort_values(by = 'created_at', ascend
          ing = False)[:10]
```

Out[21]:

		created_at
created_date	created_hour	
2020-08-22	13	90
	20	79
	18	56
	15	54
	14	46
	12	45
	19	44
	17	44
	22	43
	16	42

```
In [22]:  # 25 different languages represented in this collection
          !python twitter_lang.py ted tedtweets twitter_lang_results.csv
```

Wrote 25 to file

```
In [23]:  # Using twitter_lang.py as an example, use different fields
          !python Raya_Young_twitter_name.py ted tedtweets twitter_name_results.cs
          v
          # 2645 unique users
```

Wrote 2645 to file

```python
# Top 20 Frequency Hashtags
!python twitter_hashtags.py ted tedtweets 20
```

```
Top 20 Frequency Hashtags
TedTalk 925
TED 704
Leadership 576
Management 382
TEDx 283
tedx 265
Tedx 210
ted 160
TEDTalk 107
教育 91
Ted 77
動画 74
tedxtcet 70
فـهـد_االأحـمـدي 67
تـيـد 67
صبـاح_الـخيـر 67
talk 58
IFTHH 56
tedtalk 55
technology 52
```

```python
# What are the number of English Tweets in this collection?
dfen = df[df['lang']=='en']
len(dfen) # 3418 tweets in English
```

3418

```python
dfen[['retweet_count']].mean() # The average number of retweets in this
 collection is 28.8
max_RT = dfen.sort_values(by = 'retweet_count', ascending = False)[:1]
max_RT # The maximum number of retweets in this collection is 1235

# Sandeep Ahlawat, Lieutenant Colonel of Indian Army - https://www.youtu
be.com/watch?v=8wU-cK9G4V8
# https://twitter.com/SandyAhlawat89/status/1175335660162433027
```

	_id	created_at	text	entities	user	re
4600	5f4aa89bd6a56ce85825ae65	2020-08-21 05:44:10	RT @SandyAhlawat89: When civilian countrymen a...	{'hashtags': [], 'symbols': [], 'user_mentions...	{'id': 2880746036, 'id_str': '2880746036', 'na...	

```
In [27]: dfen[['favorite_count']].mean() # The average number of favorites in thi
         s collection is 2
         max_fave = dfen.sort_values(by = 'favorite_count', ascending = False)[:1
         ]
         max_fave # The maximum number of favorites in this collection is 818
```

Out[27]:

	_id	created_at	text	entities	user	retweet_count	fav
2920	5f4aa86f678bd9aa94f14169	2020-08-22 12:45:33	Good morning. If you haven't watched my #TedTa...	{'hashtags': [{'text': 'TedTalk', 'indices': [...	{'id': 17375057, 'id_str': '17375057', 'name':...	286	

```
# What are the shared features of the top 1000 most popular Retweets?
topRT = dfen.sort_values(by = 'retweet_count', ascending = False).head(1
000) # 1/3 of collection for training
topRT
```

	_id	created_at	text	entities	
4600	5f4aa89bd6a56ce85825ae65	2020-08-21 05:44:10	RT @SandyAhlawat89: When civilian countrymen a...	{'hashtags': [], 'symbols': [], 'user_mentions...	{'id': 2880746('id_str': '2880746('
3298	5f4aa89bd6a56ce85825a94f	2020-08-28 21:57:27	RT @AttorneyGriggs: "From Civil Rights to Soci...	{'hashtags': [{'text': 'justicefighter', 'indi...	{'id': 32931 'id_str': '329318 'nam
2821	5f4aa86f678bd9aa94f14106	2020-08-22 14:25:13	RT @hmcghee: Good morning. If you haven't watc...	{'hashtags': [{'text': 'TedTalk', 'indices': [...	{'id': 16586 'id_str': '165868 'nam
2804	5f4aa86f678bd9aa94f140f5	2020-08-22 15:14:06	RT @hmcghee: Good morning. If you haven't watc...	{'hashtags': [{'text': 'TedTalk', 'indices': [...	{'id': 1476103 'id_str': '14761032 '
2805	5f4aa86f678bd9aa94f140f6	2020-08-22 15:12:45	RT @hmcghee: Good morning. If you haven't watc...	{'hashtags': [{'text': 'TedTalk', 'indices': [...	{'id': 1454712 'id_str': '14547129 '
...	...	...	...	...	
3600	5f4aa89bd6a56ce85825aa7d	2020-08-27 03:35:21	RT @LollyDaskal: Fear and focus are the only t...	{'hashtags': [], 'symbols': [], 'user_mentions...	{'id': 198787 'id_str': '1987871 'nar
2210	5f4aa86f678bd9aa94f13ea3	2020-08-24 12:47:09	RT @LollyDaskal: Our life is a reflection of t...	{'hashtags': [{'text': 'Leadership', 'indices'...	902154554216783 'id_str': '902154
2208	5f4aa86f678bd9aa94f13ea1	2020-08-24 12:56:02	RT @LollyDaskal: Our life is a reflection of t...	{'hashtags': [{'text': 'Leadership', 'indices'...	{'id': 1403356 'id_str': '14033564 '
3080	5f4aa86f678bd9aa94f14209	2020-08-21 12:59:01	RT @LollyDaskal: A bad attitude is like a flat...	{'hashtags': [], 'symbols': [], 'user_mentions...	{'id': 1403356 'id_str': '14033564 '
2206	5f4aa86f678bd9aa94f13e9f	2020-08-24 13:09:22	RT @LollyDaskal: Our life is a reflection of t...	{'hashtags': [{'text': 'Leadership', 'indices'...	{'id': 3187094 'id_str': '3187094('

1000 rows × 10 columns

```
In [29]:   # consolidated repeated RTs
           topRT = dfen[['text', 'retweet_count']].groupby(topRT['text']).sum().sor
           t_values(by = 'retweet_count', ascending = False)
           topRT

           # Racism Has a Cost for Everyone - Heather C. McGee
           # https://www.ted.com/talks/heather_c_mcghee_racism_has_a_cost_for_every
           one?utm_source=t.co&utm_content=2020-8-20&utm_medium=referral&utm_campai
           gn=social
```

Out[29]:

text	retweet_count
RT @hmcghee: Good morning. If you haven't watched my #TedTalk, please do — and let me know what you think.	50050
RT @hmcghee: Good morning. If you haven't watched my #TedTalk, please do — and let me know what you think. https://t.co/1sOAaRf0Ng	15730
RT @maysoonzayid: TALK TO THE #DISABLED PERSON NOT THE NON DISABLED PARENTS. Thanks for coming to my #TedTalk	6810
RT @DuncanJWardle: Which is your favorite place to #ThinkDifferent? Mine is at my @TEDx talk on the Theory of #Creativity at @Tedx_AUK: htt…	2288
RT @EmergMedDr: If you get bit by an insect , it stings and hurts but you don't need to seek help straight away in an A&E.\n\nBuy some antihi…	1634
...	...
RT @LollyDaskal: A caring attitude is one of the great leadership qualities of a great leader.\n~@LollyDaskal https://t.co/ppfSHzIgez #Leade…	8
RT @LollyDaskal: You don't INSPIRE OTHERS by speaking about how amazing you are.\n\nYou INSPIRE OTHERS by showing them how amazing THEY ARE.…	8
RT @TEDxTCET: Even in these difficult times, a sister's love always shines. Here's to a quarantine Raksha Bandhan! #tedxtcet #tedxtcet2020…	8
A caring attitude is one of the great leadership qualities of a great leader.\n~@LollyDaskal https://t.co/ppfSHzIgez… https://t.co/jFj35CfF76	8
Get inspired for some family time! 😍 👇 What's your favourite #TEDtalk?\n#ece #ecd\nhttps://t.co/ObshPEhraS	8

101 rows × 1 columns

```
In [30]:   # What are the number of English Tweets in this collection?
           dfen = df[df['lang']=='en']
           len(dfen) # 3418 tweets in English
```

Out[30]:   3418

```
In [31]:  # Text Tokenization
          import nltk

          client = pymongo.MongoClient('localhost', 27017)

          db = client.ted

          db.list_collection_names()
```

Out[31]: ['tedtweets']

```
In [32]:  collection = db.tedtweets

          tweets = collection.find()

          tweetlist = list(tweets)

          textlist = [tweet['text'] for tweet in tweetlist if 'text' in tweet.keys
          ()]
          len(textlist)
```

Out[32]: 4601

```
In [33]:  all_tokens = [tok for text in textlist for tok in nltk.word_tokenize(tex
          t)]
          len(all_tokens) #119204
          all_tokens[:50]

Out[33]:  ['My',
           'college',
           'students',
           'are',
           'making',
           'their',
           'own',
           'Bitmoji',
           'classrooms',
           'and',
           'I',
           'absolutely',
           'love',
           'it',
           '!',
           '#',
           'BMCC',
           '#',
           'TED',
           'PARTY',
           'ON',
           'DUDES',
           '.',
           '#',
           'BillAndTed',
           '#',
           'bill',
           '#',
           'ted',
           '#',
           'billandtedfancam',
           '#',
           'fancam',
           '#',
           'fancams',
           '#',
           'billandtedfancams',
           'https',
           ':',
           '//t.co/5MyCwhhD0D',
           'Kolonizacja',
           'Marsa',
           'a',
           'praca',
           'w',
           'sektorze',
           'kosmicznym',
           '-',
           'moje',
           'wystąpienie']
```

```
In [34]:  textFD = nltk.FreqDist(all_tokens)
          textFD.most_common(30)
```

Out[34]: [('#', 9433),
         (':', 6521),
         ('@', 4530),
         ('https', 3466),
         ('RT', 2665),
         ('.', 2662),
         ('LollyDaskal', 1894),
         (',', 1886),
         ('you', 1604),
         ('to', 1531),
         ('a', 1243),
         ('the', 1238),
         ('and', 978),
         ('is', 956),
         ('of', 939),
         ('TedTalk', 908),
         ('//t.co/ppfSHzIgez', 824),
         (''', 800),
         ('~', 800),
         ('TED', 787),
         ('in', 635),
         ('what', 593),
         ('my', 578),
         ('for', 552),
         ('!', 549),
         ('do', 534),
         ('Leadership', 515),
         ('I', 456),
         ('de', 441),
         ('t', 431)]
```

```
In [35]: import re
         def alpha_filter(w):
             pattern = re.compile('^[^a-z]+$')
             if (pattern.match(w)):
                 return True
             else:
                 return False

         token_list = [tok for tok in all_tokens if not alpha_filter(tok)]
         token_list[:30]
```

Out[35]: ['My',
         'college',
         'students',
         'are',
         'making',
         'their',
         'own',
         'Bitmoji',
         'classrooms',
         'and',
         'absolutely',
         'love',
         'it',
         'BillAndTed',
         'bill',
         'ted',
         'billandtedfancam',
         'fancam',
         'fancams',
         'billandtedfancams',
         'https',
         '//t.co/5MyCwhhD0D',
         'Kolonizacja',
         'Marsa',
         'a',
         'praca',
         'w',
         'sektorze',
         'kosmicznym',
         'moje']

```
In [36]:  textFD = nltk.FreqDist(token_list)

          top_words = textFD.most_common(30)

          for word, freq in top_words:
              print(word, freq)
```

```
https 3466
LollyDaskal 1894
you 1604
to 1531
a 1243
the 1238
and 978
is 956
of 939
TedTalk 908
//t.co/ppfSHzIgez 824
in 635
what 593
my 578
for 552
do 534
Leadership 515
de 441
t 431
your 424
on 405
TEDx 375
If 369
know 367
Management 361
it 347
have 342
are 337
me 334
from 317
```

```
In [37]:  from nltk.tokenize import word_tokenize
          from nltk.sentiment.vader import SentimentIntensityAnalyzer

          nltk_stopwords = nltk.corpus.stopwords.words('english')
          nltk_stopwords

          # transcript analysis; word cloud, sentiment analysis
          max_views_script = max_views['Transcript'] # Do Schools Kill Creativity?
          - Sir Ken Robinson

          max_comments_script = max_comments['Transcript'] # Militant Atheism - Ri
          chard Dawkins
```

```
In [38]: views_tokens = max_views_script.apply(word_tokenize)
         views_tokens_list = [word for word in views_tokens if word not in nltk_s
         topwords]
         views_tokens_list
```

```
Out[38]: [['Good',
          'morning',
          '.',
          'How',
          'are',
          'you',
          '?',
          '(',
          'Audience',
          ')',
          'Good',
          '.',
          'It',
          "'s",
          'been',
          'great',
          ',',
          'has',
          "n't",
          'it',
          '?',
          'I',
          "'ve",
          'been',
          'blown',
          'away',
          'by',
          'the',
          'whole',
          'thing',
          '.',
          'In',
          'fact',
          ',',
          'I',
          "'m",
          'leaving',
          '.',
          '(',
          'Laughter',
          ')',
          'There',
          'have',
          'been',
          'three',
          'themes',
          'running',
          'through',
          'the',
          'conference',
          ',',
          'which',
          'are',
          'relevant',
          'to',
          'what',
          'I',
```

'want',
'to',
'talk',
'about',
'.',
'One',
'is',
'the',
'extraordinary',
'evidence',
'of',
'human',
'creativity',
'in',
'all',
'of',
'the',
'presentations',
'that',
'we',
"'ve",
'had',
'and',
'in',
'all',
'of',
'the',
'people',
'here',
';',
'just',
'the',
'variety',
'of',
'it',
'and',
'the',
'range',
'of',
'it',
'.',
'The',
'second',
'is',
'that',
'it',
"'s",
'put',
'us',
'in',
'a',
'place',
'where',
'we',
'have',
'no',
'idea',

```
'what',
"'s",
'going',
'to',
'happen',
'in',
'terms',
'of',
'the',
'future',
'.',
'No',
'idea',
'how',
'this',
'may',
'play',
'out',
'.',
'I',
'have',
'an',
'interest',
'in',
'education',
'.',
'Actually',
',',
'what',
'I',
'find',
'is',
',',
'everybody',
'has',
'an',
'interest',
'in',
'education',
'.',
'Do',
"n't",
'you',
'?',
'I',
'find',
'this',
'very',
'interesting',
'.',
'If',
'you',
"'re",
'at',
'a',
'dinner',
'party',
```

',',
'and',
'you',
'say',
'you',
'work',
'in',
'education',
'—',
'actually',
',',
'you',
"'re",
'not',
'often',
'at',
'dinner',
'parties',
',',
'frankly',
'.',
'(',
'Laughter',
')',
'If',
'you',
'work',
'in',
'education',
',',
'you',
"'re",
'not',
'asked',
'.',
'(',
'Laughter',
')',
'And',
'you',
"'re",
'never',
'asked',
'back',
',',
'curiously',
'.',
'That',
"'s",
'strange',
'to',
'me',
'.',
'But',
'if',
'you',
'are',

```
',',
'and',
'you',
'say',
'to',
'somebody',
',',
'you',
'know',
',',
'they',
'say',
',',
'``',
'What',
'do',
'you',
'do',
'?',
"''",
'and',
'you',
'say',
'you',
'work',
'in',
'education',
',',
'you',
'can',
'see',
'the',
'blood',
'run',
'from',
'their',
'face',
'.',
'They',
"'re",
'like',
',',
'``',
'Oh',
'my',
'God',
'.',
'Why',
'me',
'?',
"''",
'(',
'Laughter',
')',
'``',
'My',
'one',
```

'night',
'out',
'all',
'week',
'.',
"''",
'(',
'Laughter',
')',
'But',
'if',
'you',
'ask',
'about',
'their',
'education',
',',
'they',
'pin',
'you',
'to',
'the',
'wall',
',',
'because',
'it',
"'s",
'one',
'of',
'those',
'things',
'that',
'goes',
'deep',
'with',
'people',
',',
'am',
'I',
'right',
'?',
'Like',
'religion',
'and',
'money',
'and',
'other',
'things',
'.',
'So',
'I',
'have',
'a',
'big',
'interest',
'in',
'education',

```
',',
'and',
'I',
'think',
'we',
'all',
'do',
'.',
'We',
'have',
'a',
'huge',
'vested',
'interest',
'in',
'it',
',',
'partly',
'because',
'it',
"'s",
'education',
'that',
"'s",
'meant',
'to',
'take',
'us',
'into',
'this',
'future',
'that',
'we',
'ca',
"n't",
'grasp',
'.',
'If',
'you',
'think',
'of',
'it',
',',
'children',
'starting',
'school',
'this',
'year',
'will',
'be',
'retiring',
'in',
'2065',
'.',
'Nobody',
'has',
'a',
```

```
'clue',
',',
'despite',
'all',
'the',
'expertise',
'that',
"'s",
'been',
'on',
'parade',
'for',
'the',
'past',
'four',
'days',
',',
'what',
'the',
'world',
'will',
'look',
'like',
'in',
'five',
'years',
"'",
'time',
'.',
'And',
'yet',
',',
'we',
"'re",
'meant',
'to',
'be',
'educating',
'them',
'for',
'it',
'.',
'So',
'the',
'unpredictability',
',',
'I',
'think',
',',
'is',
'extraordinary',
'.',
'And',
'the',
'third',
'part',
'of',
```

```
'this',
'is',
'that',
'we',
"'ve",
'all',
'agreed',
',',
'nonetheless',
',',
'on',
'the',
'really',
'extraordinary',
'capacities',
'that',
'children',
'have',
'—',
'their',
'capacities',
'for',
'innovation',
'.',
'I',
'mean',
',',
'Sirena',
'last',
'night',
'was',
'a',
'marvel',
',',
'was',
"n't",
'she',
'?',
'Just',
'seeing',
'what',
'she',
'could',
'do',
'.',
'And',
'she',
"'s",
'exceptional',
',',
'but',
'I',
'think',
'she',
"'s",
'not',
',',
```

```
'so',
'to',
'speak',
',',
'exceptional',
'in',
'the',
'whole',
'of',
'childhood',
'.',
'What',
'you',
'have',
'there',
'is',
'a',
'person',
'of',
'extraordinary',
'dedication',
'who',
'found',
'a',
'talent',
'.',
'And',
'my',
'contention',
'is',
',',
'all',
'kids',
'have',
'tremendous',
'talents',
',',
'and',
'we',
'squander',
'them',
',',
'pretty',
'ruthlessly',
'.',
'So',
'I',
'want',
'to',
'talk',
'about',
'education',
',',
'and',
'I',
'want',
'to',
```

'talk',
'about',
'creativity',
'.',
'My',
'contention',
'is',
'that',
'creativity',
'now',
'is',
'as',
'important',
'in',
'education',
'as',
'literacy',
',',
'and',
'we',
'should',
'treat',
'it',
'with',
'the',
'same',
'status',
'.',
'(',
'Applause',
')',
'Thank',
'you',
'.',
'(',
'Applause',
')',
'That',
'was',
'it',
',',
'by',
'the',
'way',
'.',
'Thank',
'you',
'very',
'much',
'.',
'(',
'Laughter',
')',
'So',
',',
'15',
'minutes',

```
'left',
'.',
'(',
'Laughter',
')',
'``',
'Well',
',',
'I',
'was',
'born',
'...',
'``',
'(',
'Laughter',
')',
'I',
'heard',
'a',
'great',
'story',
'recently',
'—',
'I',
'love',
'telling',
'it',
'—',
'of',
'a',
'little',
'girl',
'who',
'was',
'in',
'a',
'drawing',
'lesson',
'.',
'She',
'was',
'six',
',',
'and',
'she',
'was',
'at',
'the',
'back',
',',
'drawing',
',',
'and',
'the',
'teacher',
'said',
'this',
```

'girl',
'hardly',
'ever',
'paid',
'attention',
',',
'and',
'in',
'this',
'drawing',
'lesson',
',',
'she',
'did',
'.',
'The',
'teacher',
'was',
'fascinated',
'.',
'She',
'went',
'over',
'to',
'her',
',',
'and',
'she',
'said',
',',
'``',
'What',
'are',
'you',
'drawing',
'?',
"''",
'And',
'the',
'girl',
'said',
',',
'``',
'I',
"'m",
'drawing',
'a',
'picture',
'of',
'God',
'.',
"''",
'And',
'the',
'teacher',
'said',
',',

```
'``',
'But',
'nobody',
'knows',
'what',
'God',
'looks',
'like',
'.',
"''",
'And',
'the',
'girl',
'said',
',',
'``',
'They',
'will',
'in',
'a',
'minute',
'.',
"''",
'(',
'Laughter',
')',
'When',
'my',
'son',
'was',
'four',
'in',
'England',
'—',
'actually',
',',
'he',
'was',
'four',
'everywhere',
',',
'to',
'be',
'honest',
'.',
'(',
'Laughter',
')',
'If',
'we',
"'re",
'being',
'strict',
'about',
'it',
',',
'wherever',
```

'he',
'went',
',',
'he',
'was',
'four',
'that',
'year',
'.',
'He',
'was',
'in',
'the',
'Nativity',
'play',
'.',
'Do',
'you',
'remember',
'the',
'story',
'?',
'(',
'Laughter',
')',
'No',
',',
'it',
'was',
'big',
',',
'it',
'was',
'a',
'big',
'story',
'.',
'Mel',
'Gibson',
'did',
'the',
'sequel',
',',
'you',
'may',
'have',
'seen',
'it',
'.',
'(',
'Laughter',
')',
'``',
'Nativity',
'II',
'.',
"''",

'But',
'James',
'got',
'the',
'part',
'of',
'Joseph',
',',
'which',
'we',
'were',
'thrilled',
'about',
'.',
'We',
'considered',
'this',
'to',
'be',
'one',
'of',
'the',
'lead',
'parts',
'.',
'We',
'had',
'the',
'place',
'crammed',
'full',
'of',
'agents',
'in',
'T-shirts',
':',
'``',
'James',
'Robinson',
'IS',
'Joseph',
'!',
"''",
'(',
'Laughter',
')',
'He',
'did',
"n't",
'have',
'to',
'speak',
',',
'but',
'you',
'know',
'the',

```
'bit',
'where',
'the',
'three',
'kings',
'come',
'in',
'?',
'They',
'come',
'in',
'bearing',
'gifts',
',',
'gold',
',',
'frankincense',
'and',
'myrrh',
'.',
'This',
'really',
'happened',
'.',
'We',
'were',
'sitting',
'there',
',',
'and',
'I',
'think',
'they',
'just',
'went',
'out',
'of',
'sequence',
',',
'because',
'we',
'talked',
'to',
'the',
'little',
'boy',
'afterward',
'and',
'said',
',',
'``',
'You',
'OK',
'with',
'that',
'?',
"''",
```

```
'They',
'said',
',',
'``',
'Yeah',
',',
'why',
'?',
'Was',
'that',
'wrong',
'?',
"''",
'They',
'just',
'switched',
'.',
'The',
'three',
'boys',
'came',
'in',
',',
'four-year-olds',
'with',
'tea',
'towels',
'on',
'their',
'heads',
'.',
...]]
```

```
In [39]:  from textblob import TextBlob

          text = str(views_tokens_list)

          blob = TextBlob(text)
          blob.tags             # [('The', 'DT'), ('titular', 'JJ'),
                                #  ('threat', 'NN'), ('of', 'IN'), ...]

          blob.noun_phrases     # WordList(['titular threat', 'blob',
                                #           'ultimate movie monster',
                                #           'amoeba-like mass', ...])

          for sentence in blob.sentences:
              print(sentence.sentiment.polarity)
```

0.7
0.0
0.7
0.8
0.2
0.0
0.4
0.16666666666666666
0.0
0.0
0.0
0.0
0.0
0.65
0.0
0.0
-0.05
-0.05
0.0
0.0
0.0
0.0
0.0
0.14285714285714285
-0.125
0.0
0.20000000000000004
0.0
-0.25
0.0
0.3333333333333333
0.16666666666666666
-0.15625
0.0
0.5111111111111111
0.3333333333333333
-0.13888888888888892
0.0
0.2
0.0
0.0
0.26
0.0
0.37083333333333335
-0.14583333333333334
0.0
0.0
0.0
0.0
0.0
0.3
0.0
0.0
0.0
0.0
0.0
0.0

0.6

0.0

0.4375

0.0

0.0

0.2

0.15625

0.0

−0.5

0.0

0.0

0.04722222222222221

0.0

0.0

−0.3

0.0

0.2857142857142857

−0.5

−0.078125

−0.20833333333333334

0.5

−0.5

0.0

0.0

−1.0

0.5

0.0

0.0

−0.05

0.0

0.0

0.0

0.0

0.1

0.0

0.13636363636363635

0.0

0.0

0.0

0.0

0.0

0.0

−0.3125

0.0

−0.8

−0.1

0.0

0.0

−0.19444444444444448

0.0

−0.3

0.0

0.0

0.0

0.0

0.7

0.5

0.0
-0.025
0.2
0.0
0.3333333333333333
0.0
0.0
0.0
0.0
0.5
0.0
0.0
0.225
0.2
0.0
0.0
0.0
0.4
0.52
0.0
0.0
0.0
0.5
0.0
-0.16666666666666666
-0.125
0.3
0.0
0.5
0.0
0.0
0.0
0.0
-0.1
-0.1
-0.015151515151515152
-0.015151515151515152
0.6
0.0
0.0
0.0
0.1
0.0
0.0
0.0
0.0
0.0
0.1
0.0
0.0
0.43333333333333335
0.6
0.2857142857142857
0.0
0.08333333333333333
0.2
0.06666666666666667

0.1
0.5499999999999999
0.0
0.25
0.5
0.4000000000000001
0.15
0.35
0.0
0.0
0.0
-0.14583333333333331
0.0
0.2
0.0
0.0
0.0
0.0
0.2
0.0
0.0
0.5
0.0
0.2916666666666667
0.0
0.0
0.25
0.0
0.0
0.0
0.7
0.0
-0.4
0.0
0.0
0.03333333333333333
0.1
0.10000000000000002
0.15
0.13636363636363635
0.0
0.5666666666666667
0.0
0.0
0.0
0.0
1.0
0.0
0.0
0.0
0.5
0.2
0.0
0.0
0.0
0.0
0.0

0.4
0.25
0.0
-0.4
-0.1875
0.0
0.0
0.0
0.0
-0.0325
0.0
0.0
0.0
-0.2
-0.7142857142857143
0.0
0.0
0.0
0.0
1.0
0.35
0.0
0.18333333333333335
0.0
1.0
0.0
0.3625
0.07222222222222223
-0.125
0.02727272727272727
0.16666666666666666
0.0
0.0
1.0
0.0
0.2857142857142857
0.0
0.3
0.25
0.1
0.0
0.0
0.26
0.0

```
In [40]:  comments_tokens = max_comments_script.apply(word_tokenize)
          comments_tokens_list = [word for word in comments_tokens if word not in
          nltk_stopwords]
          comments_tokens_list
```

```
Out[40]: [['That',
  'splendid',
  'music',
  ',',
  'the',
  'coming-in',
  'music',
  ',',
  '``',
  'The',
  'Elephant',
  'March',
  "''",
  'from',
  '``',
  'Aida',
  ',',
  "''",
  'is',
  'the',
  'music',
  'I',
  "'ve",
  'chosen',
  'for',
  'my',
  'funeral',
  '.',
  '(',
  'Laughter',
  ')',
  'And',
  'you',
  'can',
  'see',
  'why',
  '.',
  'It',
  "'s",
  'triumphal',
  '.',
  'I',
  'wo',
  "n't",
  'feel',
  'anything',
  ',',
  'but',
  'if',
  'I',
  'could',
  ',',
  'I',
  'would',
  'feel',
  'triumphal',
  'at',
```

```
'having',
'lived',
'at',
'all',
',',
'and',
'at',
'having',
'lived',
'on',
'this',
'splendid',
'planet',
',',
'and',
'having',
'been',
'given',
'the',
'opportunity',
'to',
'understand',
'something',
'about',
'why',
'I',
'was',
'here',
'in',
'the',
'first',
'place',
',',
'before',
'not',
'being',
'here',
'.',
'Can',
'you',
'understand',
'my',
'quaint',
'English',
'accent',
'?',
'(',
'Laughter',
')',
'Like',
'everybody',
'else',
',',
'I',
'was',
'entranced',
'yesterday',
```

'by',
'the',
'animal',
'session',
'.',
'Robert',
'Full',
'and',
'Frans',
'Lanting',
'and',
'others',
';',
'the',
'beauty',
'of',
'the',
'things',
'that',
'they',
'showed',
'.',
'The',
'only',
'slight',
'jarring',
'note',
'was',
'when',
'Jeffrey',
'Katzenberg',
'said',
'of',
'the',
'mustang',
',',
'``',
'the',
'most',
'splendid',
'creatures',
'that',
'God',
'put',
'on',
'this',
'earth',
'.',
"''",
'Now',
'of',
'course',
',',
'we',
'know',
'that',
'he',

```
'did',
"n't",
'really',
'mean',
'that',
',',
'but',
'in',
'this',
'country',
'at',
'the',
'moment',
',',
'you',
'ca',
"n't",
'be',
'too',
'careful',
'.',
'(',
'Laughter',
')',
'I',
"'m",
'a',
'biologist',
',',
'and',
'the',
'central',
'theorem',
'of',
'our',
'subject',
':',
'the',
'theory',
'of',
'design',
',',
'Darwin',
"'s",
'theory',
'of',
'evolution',
'by',
'natural',
'selection',
'.',
'In',
'professional',
'circles',
'everywhere',
',',
'it',
```

```
"'s",
'of',
'course',
'universally',
'accepted',
'.',
'In',
'non-professional',
'circles',
'outside',
'America',
',',
'it',
"'s",
'largely',
'ignored',
'.',
'But',
'in',
'non-professional',
'circles',
'within',
'America',
',',
'it',
'arouses',
'so',
'much',
'hostility',
'—',
'(',
'Laughter',
')',
'it',
"'s",
'fair',
'to',
'say',
'that',
'American',
'biologists',
'are',
'in',
'a',
'state',
'of',
'war',
'.',
'The',
'war',
'is',
'so',
'worrying',
'at',
'present',
',',
'with',
```

```
'court',
'cases',
'coming',
'up',
'in',
'one',
'state',
'after',
'another',
',',
'that',
'I',
'felt',
'I',
'had',
'to',
'say',
'something',
'about',
'it',
'.',
'If',
'you',
'want',
'to',
'know',
'what',
'I',
'have',
'to',
'say',
'about',
'Darwinism',
'itself',
',',
'I',
"'m",
'afraid',
'you',
"'re",
'going',
'to',
'have',
'to',
'look',
'at',
'my',
'books',
',',
'which',
'you',
'wo',
"n't",
'find',
'in',
'the',
'bookstore',
```

```
'outside',
'.',
'(',
'Laughter',
')',
'Contemporary',
'court',
'cases',
'often',
'concern',
'an',
'allegedly',
'new',
'version',
'of',
'creationism',
',',
'called',
'``',
'Intelligent',
'Design',
',',
"''",
'or',
'ID',
'.',
'Do',
"n't",
'be',
'fooled',
'.',
'There',
"'s",
'nothing',
'new',
'about',
'ID',
'.',
'It',
"'s",
'just',
'creationism',
'under',
'another',
'name',
',',
'rechristened',
'—',
'I',
'choose',
'the',
'word',
'advisedly',
'—',
'(',
'Laughter',
')',
```

'for',
'tactical',
',',
'political',
'reasons',
'.',
'The',
'arguments',
'of',
'so-called',
'ID',
'theorists',
'are',
'the',
'same',
'old',
'arguments',
'that',
'had',
'been',
'refuted',
'again',
'and',
'again',
',',
'since',
'Darwin',
'down',
'to',
'the',
'present',
'day',
'.',
'There',
'is',
'an',
'effective',
'evolution',
'lobby',
'coordinating',
'the',
'fight',
'on',
'behalf',
'of',
'science',
',',
'and',
'I',
'try',
'to',
'do',
'all',
'I',
'can',
'to',
'help',

```
'them',
',',
'but',
'they',
'get',
'quite',
'upset',
'when',
'people',
'like',
'me',
'dare',
'to',
'mention',
'that',
'we',
'happen',
'to',
'be',
'atheists',
'as',
'well',
'as',
'evolutionists',
'.',
'They',
'see',
'us',
'as',
'rocking',
'the',
'boat',
',',
'and',
'you',
'can',
'understand',
'why',
'.',
'Creationists',
',',
'lacking',
'any',
'coherent',
'scientific',
'argument',
'for',
'their',
'case',
',',
'fall',
'back',
'on',
'the',
'popular',
'phobia',
'against',
```

'atheism',
':',
'Teach',
'your',
'children',
'evolution',
'in',
'biology',
'class',
',',
'and',
'they',
"'ll",
'soon',
'move',
'on',
'to',
'drugs',
',',
'grand',
'larceny',
'and',
'sexual',
'``',
'pre-version',
'.',
"''",
'(',
'Laughter',
')',
'In',
'fact',
',',
'of',
'course',
',',
'educated',
'theologians',
'from',
'the',
'Pope',
'down',
'are',
'firm',
'in',
'their',
'support',
'of',
'evolution',
'.',
'This',
'book',
',',
'``',
'Finding',
'Darwin',
"'s",

'God',
',',
"''",
'by',
'Kenneth',
'Miller',
',',
'is',
'one',
'of',
'the',
'most',
'effective',
'attacks',
'on',
'Intelligent',
'Design',
'that',
'I',
'know',
'and',
'it',
"'s",
'all',
'the',
'more',
'effective',
'because',
'it',
"'s",
'written',
'by',
'a',
'devout',
'Christian',
'.',
'People',
'like',
'Kenneth',
'Miller',
'could',
'be',
'called',
'a',
'``',
'godsend',
"''",
'to',
'the',
'evolution',
'lobby',
',',
'(',
'Laughter',
')',
'because',
'they',

```
'expose',
'the',
'lie',
'that',
'evolutionism',
'is',
',',
'as',
'a',
'matter',
'of',
'fact',
',',
'tantamount',
'to',
'atheism',
'.',
'People',
'like',
'me',
',',
'on',
'the',
'other',
'hand',
',',
'rock',
'the',
'boat',
'.',
'But',
'here',
',',
'I',
'want',
'to',
'say',
'something',
'nice',
'about',
'creationists',
'.',
'It',
"'s",
'not',
'a',
'thing',
'I',
'often',
'do',
',',
'so',
'listen',
'carefully',
'.',
'(',
'Laughter',
```

```
')',
'I',
'think',
'they',
"'re",
'right',
'about',
'one',
'thing',
'.',
'I',
'think',
'they',
"'re",
'right',
'that',
'evolution',
'is',
'fundamentally',
'hostile',
'to',
'religion',
'.',
'I',
"'ve",
'already',
'said',
'that',
'many',
'individual',
'evolutionists',
',',
'like',
'the',
'Pope',
',',
'are',
'also',
'religious',
',',
'but',
'I',
'think',
'they',
"'re",
'deluding',
'themselves',
'.',
'I',
'believe',
'a',
'true',
'understanding',
'of',
'Darwinism',
'is',
'deeply',
```

'corrosive',
'to',
'religious',
'faith',
'.',
'Now',
',',
'it',
'may',
'sound',
'as',
'though',
'I',
"'m",
'about',
'to',
'preach',
'atheism',
',',
'and',
'I',
'want',
'to',
'reassure',
'you',
'that',
'that',
"'s",
'not',
'what',
'I',
"'m",
'going',
'to',
'do',
'.',
'In',
'an',
'audience',
'as',
'sophisticated',
'as',
'this',
'one',
',',
'that',
'would',
'be',
'preaching',
'to',
'the',
'choir',
'.',
'No',
',',
'what',
'I',

'want',
'to',
'urge',
'upon',
'you',
'—',
'(',
'Laughter',
')',
'Instead',
',',
'what',
'I',
'want',
'to',
'urge',
'upon',
'you',
'is',
'militant',
'atheism',
'.',
'(',
'Laughter',
')',
'(',
'Applause',
')',
'But',
'that',
"'s",
'putting',
'it',
'too',
'negatively',
'.',
'If',
'I',
'was',
'a',
'person',
'who',
'were',
'interested',
'in',
'preserving',
'religious',
'faith',
',',
'I',
'would',
'be',
'very',
'afraid',
'of',
'the',
'positive',

'power',
'of',
'evolutionary',
'science',
',',
'and',
'indeed',
'science',
'generally',
',',
'but',
'evolution',
'in',
'particular',
',',
'to',
'inspire',
'and',
'enthrall',
',',
'precisely',
'because',
'it',
'is',
'atheistic',
'.',
'Now',
',',
'the',
'difficult',
'problem',
'for',
'any',
'theory',
'of',
'biological',
'design',
'is',
'to',
'explain',
'the',
'massive',
'statistical',
'improbability',
'of',
'living',
'things',
'.',
'Statistical',
'improbability',
'in',
'the',
'direction',
'of',
'good',
'design',
'—',

```
'```',
'complexity',
"''",
'is',
'another',
'word',
'for',
'this',
'.',
'The',
'standard',
'creationist',
'argument',
'—',
'there',
'is',
'only',
'one',
';',
'they',
"'re",
'all',
'reduced',
'to',
'this',
'one',
'—',
'takes',
'off',
'from',
'a',
'statistical',
'improbability',
'.',
'Living',
'creatures',
'are',
'too',
'complex',
'to',
'have',
'come',
'about',
'by',
'chance',
';',
'therefore',
',',
'they',
'must',
'have',
'had',
'a',
'designer',
'.',
'This',
'argument',
```

```
    'of',
    'course',
    ',',
    'shoots',
    'itself',
    'in',
    'the',
    'foot',
    '.',
    'Any',
    'designer',
    'capable',
    'of',
    'designing',
    'something',
    'really',
    'complex',
    'has',
    'to',
    'be',
    'even',
    'more',
    'complex',
    'himself',
    ',',
    'and',
    'that',
    "'s",
    'before',
    'we',
    'even',
    ...]]
```

```
In [41]: text = str(comments_tokens_list)

         blob = TextBlob(text)
         blob.tags              # [('The', 'DT'), ('titular', 'JJ'),
                                #  ('threat', 'NN'), ('of', 'IN'), ...]

         blob.noun_phrases      # WordList(['titular threat', 'blob',
                                #           'ultimate movie monster',
                                #           'amoeba-like mass', ...])

         for sentence in blob.sentences:
             print(sentence.sentiment.polarity)
```

0.8333333333333334
0.0
0.0
0.5416666666666667
0.0
0.0
0.35
0.2916666666666667
-0.20625
-0.022222222222222216
0.05
0.10714285714285714
0.3
0.0
-0.3
0.36767676767676766
0.0
0.13636363636363635
0.0
-0.013888888888888895
0.6
0.0
0.42000000000000004
-0.1777777777777778
0.5000000000000001
0.0
-0.125
0.6
-0.1
0.2857142857142857
0.2857142857142857
0.16666666666666666
0.11666666666666665
0.4
0.5
0.0
-0.3
0.04484848484848485
-0.25
0.7
0.0
-0.3
0.0
-0.020833333333333333
0.06666666666666667
0.3
0.2
0.3
0.85
-0.6999999999999998
0.0
0.2857142857142857
0.2
0.04107142857142857
0.0
0.0
0.7

0.0
-0.25
0.0
0.21428571428571427
-0.6999999999999998
0.0
0.0
0.0
1.0
0.15000000000000002
0.0
0.2
-0.0625
0.0
0.0
0.4666666666666666
0.0
0.5
-0.16666666666666666
0.35714285714285715
0.016666666666666663
-0.25
0.0
0.07777777777777777
-0.1
-0.1
0.06666666666666665
0.0
0.0
0.0
0.0
-0.5
0.0
0.0
0.0
0.0
0.2
0.5
-0.25
0.0
0.0
-0.25
0.0
0.15555555555555556
0.5
0.0
0.3666666666666667
0.6
0.0
0.04999999999999999
-0.15454545454545454
0.10000000000000002
0.0
0.6
0.05
0.125
0.05

0.03333333333333333
0.0
0.19999999999999998
0.2
0.0
-0.024242424242424242
-0.15555555555555559
0.0
0.0
0.10400000000000001
0.125
0.06666666666666668
0.4375
0.078125
0.28571428571428575
0.0
0.7
0.0
0.0
0.0
-0.006249999999999999
0.019999999999999997
0.03333333333333334
0.4242857142857143
0.24000000000000005
0.0
0.0
0.9
0.0
0.0
-0.06666666666666667
-0.06666666666666667
0.16666666666666666
0.3068181818181818
0.0
-0.0625
0.0
0.3
0.1375
0.0
0.0
0.275
0.0
0.06666666666666665
0.08666666666666667
0.0
0.0
0.0
0.16666666666666666
-0.3
0.11666666666666665
0.1
0.0022222222222222088
0.5
0.0
0.45
-0.09166666666666665

0.0
0.0
0.0
0.0
-0.3125
0.0
0.16666666666666666
0.0
0.0
0.0
0.5
0.25
0.0
0.0
0.025
-0.03333333333333333
-0.21666666666666667
0.0
0.4666666666666666
0.2888888888888889
0.1
0.0
-0.5
-0.19999999999999998
-0.075
0.25
-0.146
0.32187499999999997
0.048958333333333326
0.23333333333333334
0.0
-0.1875
0.08363997113997117
0.0
0.3
0.16666666666666666
0.5
0.375
0.0
0.0
0.0
0.0
0.0
0.5
0.26
0.0