Exporatory Data Analysis

Exploratory data analysis of Non-HN dataset.

NB.: using `maths` conda environment of `@paultsw` 's computer.

In [9]:
```
# import the usual suspects for EDA:
import numpy as np
import pandas as pd
import scipy.stats as stats
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
# system utils:
import csv
import sys
import os
from tqdm import tqdm, trange
```

In [3]:
```
# sklearn tools:
#from sklearn import (...)
```

In [8]:
```
# gensim:
from gensim.test.utils import common_texts, get_tmpfile
from gensim.models import Word2Vec, FastText
```

I. import dataset and do some basic checks.

In [2]:
```
data_df = pd.read_csv("../data/train_eng_labeled_01_19.csv", index_col=False)
data_df.head(10)
```

Out[2]:

	title	url	noneng
0	google cloud sql for postgres now in beta	google	0
1	drivetribes modern take on cqrs with apache flink	data-artisans	0
2	vmware joins linux foundation and what about t...	gnumonks	0
3	running spark sql cern queries 5x faster on sn...	snappydata	0
4	sjcl stanford javascript crypto library	github	0
5	zcoin implementation bug enabled attacker to c...	wordpress	0
6	show hn scribblechat for ios opengl chat clien...	getscribblechat	0
7	a tale of three kings testing python elixir a...	medium	0
8	the selinux coloring book pdf	redhat	0
9	announcing docker enterprise edition	docker	0

In [13]:
```
count_noneng = data_df[data_df['noneng'] == 1].shape[0]
count_eng = data_df[data_df['noneng'] == 0].shape[0]
print("Num. non-eng: {}".format(count_noneng))
print("Num. eng: {}".format(count_eng))
print("Ratio (Non-Eng : Eng) = {}%".format(100. * count_noneng / float(count_eng)))
```

```
Num. non-eng: 333
Num. eng: 21986
Ratio (Non-Eng : Eng) = 1.51460020013%
```

II. examine some word statistics and generate a pool of unique words.

In [15]:
```
# [TODO: THIS SECTION IS WORK-IN-PROGRESS]
for sentence in list(data_df[data_df['noneng'] == 1]['title'].values):

```

Out[15]:
```
['experience i spent 29 years in solitary confinement 2010',
 'fancy trees with botanist',
 'how propagandists abuse the internet and manipulate the public pdf',
 'the hidden cost of privatization',
 'a 12-month campaign of fake news to influence elections costs 400000',
 'berkeleys attack on housing',
 'louisiana records give insight into businesses that utilize prison labor',
 'e-cigarettes potentially as harmful as tobacco cigarettes',
 'a radical new hypothesis in medicine give patients drugs they know dont work',
 'citybikes bike sharing networks around the world',
 'tv batman actor adam west dies at 88',
 'optimizing things in the ussr',
 'doing things the wrong way to get the right result',
 'the multibillion euro theft',
 'i need loyalty james comeys riveting prepared testimony annotated',
 'even moderate drinking could harm the brain',
 'oldest fossils of homo sapiens found in morocco',
 'five men agree to stand directly under an exploding nuclear bomb',
 'no one gives a fuck about climate change',
 'las crisis high rents low pay and 2000 doesnt buy much',
 'stop trying to be original and be prolific instead',
 'trump announces plans to privatize us air traffic control system',
 'best answer to sell me this pen i have ever seen',
 'saudi arabia bahrain egypt uae cut diplomatic ties with qatar',
 'london bridge multiple casualties after incident',
 'no your phone didnt ring so why voice mail from a telemarketer',
 'all back issues of omni magazine now available online',
 'wal-mart will pay employees to deliver packages on their way home',
 'trump quits paris climate accord calling it a bad deal for the us',
 'us quits paris climate pact',
 'trump is pulling the us out of the paris climate agreement',
 'science was ignored today',
 'trump to withdraw from paris accord',
 'paris climate agreement trump withdraws us from global accord',
 'will trumps slow-mo walkaway finally provoke consequences for planetary arson',
 'trump just withdrew the us from the paris climate agreement',
 'trump will start years-long process to withdraw from paris climate agreement',
 'the swiss leaks and panama papers open a window on the tax dodgers world'
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

III. exploration of word embedding models: Word2Vec, FastText

In [19]:
```
# get formatted list of lists of words from all titles:
all_sentences = [s.strip().split() for s in data_df['title'].values]
# ... from just the engineering titles:
eng_sentences = [s.strip().split() for s in data_df[data_df['noneng'] == 0]['title'].valu
es]
# ... and from just the non-engineering titles:
noneng_sentences = [s.strip().split() for s in data_df[data_df['noneng'] == 1]['title'].v
alues]
```

In [21]:
```
# train a word2vec model on all sentences:
# arguments:
# size ~ dimension of embedding vector space
# window ~ window size of each word
# min_count ~ ignore words that appear fewer than min_count times in the entire corpus of data
# workers ~ threads to use
# sg ~ if 1, use skip-gram training (0 -> continuous bag-of-words model; we're using skip-gram for rare words)
model_w2v = Word2Vec(sentences=all_sentences, size=100, window=5, min_count=5, workers=3, sg=1)
```

In [25]:
```
# check out a few most-similar-words evaluations on some random words and compute distances:
model_w2v.wv.most_similar("campaign")
```

Out[25]:
```
[('safety', 0.9989723563194275),
 ('releasing', 0.9985957145690918),
 ('contains', 0.9985491037368774),
 ('readers', 0.9985160827636719),
 ('next-gen', 0.9984472990036011),
 ('eu', 0.9984416365623474),
 ('girl', 0.9984038472175598),
 ('xss', 0.9983392953872681),
 ('collects', 0.9983255863189697),
 ('picks', 0.9982678890228271)]
```

In [33]:
```
[np.sum(np.power(model_w2v.wv['campaign'] - model_w2v.wv[_word], 2.)) for _word in ['safety', 'contains', 'next-gen']]
```

Out[33]:
```
[0.024290895, 0.06838648, 0.04055692]
```

In [26]:
```
model_w2v.wv.most_similar("internet")
```

Out[26]:
```
[('happened', 0.9939441084861755),
 ('almost', 0.9933430552482605),
 ('cost', 0.9920545816421509),
 ('replace', 0.9913919568061829),
 ('broken', 0.9913278818130493),
 ('startups', 0.9911482334136963),
 ('employees', 0.9910067915916443),
 ('america', 0.99090576171875),
 ('his', 0.9904655814170837),
 ('genius', 0.9904117584228516)]
```

In [34]:
```
[np.sum(np.power(model_w2v.wv['internet'] - model_w2v.wv[_word], 2.)) for _word in ['happ
ened', 'broken', 'genius']]
```

Out[34]:
```
[0.028187377, 0.04174581, 0.040919397]
```

In [27]:
```
model_w2v.wv.most_similar("right")
```

Out[27]:
```
[('slower', 0.9909306764602661),
 ('long', 0.9904853105545044),
 ('fall', 0.9897018074989319),
 ('safer', 0.9895232915878296),
 ('2007', 0.989295482635498),
 ('worth', 0.9891735315322876),
 ('cobol', 0.988878607749939),
 ('cpython', 0.9883400797843933),
 ('solve', 0.9883016347885132),
 ('easiest', 0.9879031777381897)]
```

In [35]:
```
[np.sum(np.power(model_w2v.wv['right'] - model_w2v.wv[_word], 2.)) for _word in ['slower'
, 'worth', 'cobol']]
```

Out[35]:
```
[0.0458754, 0.047672693, 0.19075617]
```

In [28]:
```
model_w2v.wv.most_similar("costs")
```

Out[28]:
```
[('lawsuit', 0.9985125064849854),
 ('sun', 0.9983959197998047),
 ('december', 0.998370349407196),
 ('patches', 0.9980621337890625),
 ('drupal', 0.9978747367858887),
 ('paris', 0.9977258443832397),
 ('sequel', 0.9976789355278015),
 ('rights', 0.9974321126937866),
 ('shares', 0.9973852634429932),
 ('follow', 0.9973822832107544)]
```

In [36]:
```
[np.sum(np.power(model_w2v.wv['costs'] - model_w2v.wv[_word], 2.)) for _word in ['lawsuit
', 'paris', 'shares']]
```

Out[36]:
```
[0.0120189395, 0.06709887, 0.064355806]
```

In [29]:
```
model_w2v.wv.most_similar("electric")
```

Out[29]:
```
[('illegal', 0.9990853071212769),
 ('radios', 0.9990560412406921),
 ('bubble', 0.9990074634552002),
 ('acquisition', 0.9990014433860779),
 ('japan', 0.9989989995956421),
 ('strange', 0.9989743828773499),
 ('39', 0.9989455342292786),
 ('accused', 0.9989310503005981),
 ('500m', 0.9989277124404907),
 ('spacex', 0.9989216923713684)]
```

In [37]:
```
[np.sum(np.power(model_w2v.wv['electric'] - model_w2v.wv[_word], 2.)) for _word in ['ille
gal', 'strange', 'spacex']]
```

Out[37]:
```
[0.0027745592, 0.004293309, 0.011509068]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: