Kshitija Shelar   **Text Analytics COMP47 60**   Date: 25/10/20

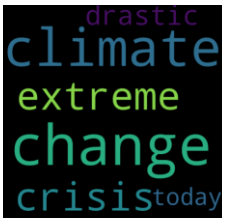
20200155   Practical 4

1. A.

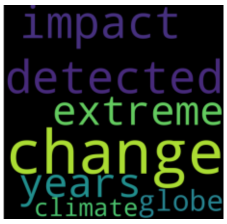
Original text document (1/10)	After removing stop words
The artic is melting ice capades are melting at a very high rate causing rise in water level	Artic, melting, ice, capades, melting, high, rate, causing, rise, water, level

1. B. Word cloud after performing TF calculations for every document.

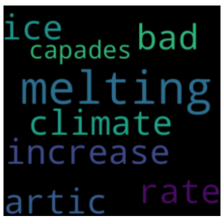
Doc1



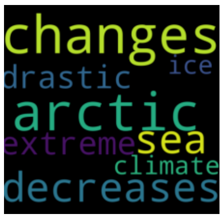
Doc2



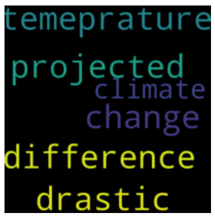
Doc3



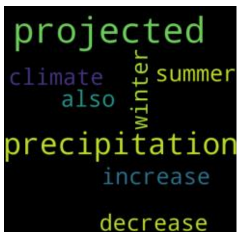
Doc4



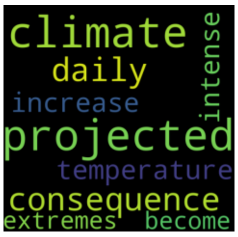
Doc5



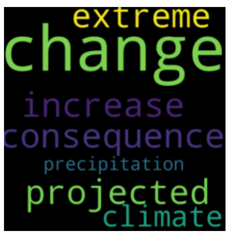
Doc6



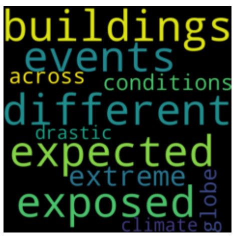
Doc7



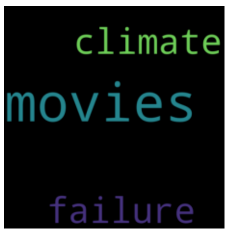
Doc8



Doc9



Doc10

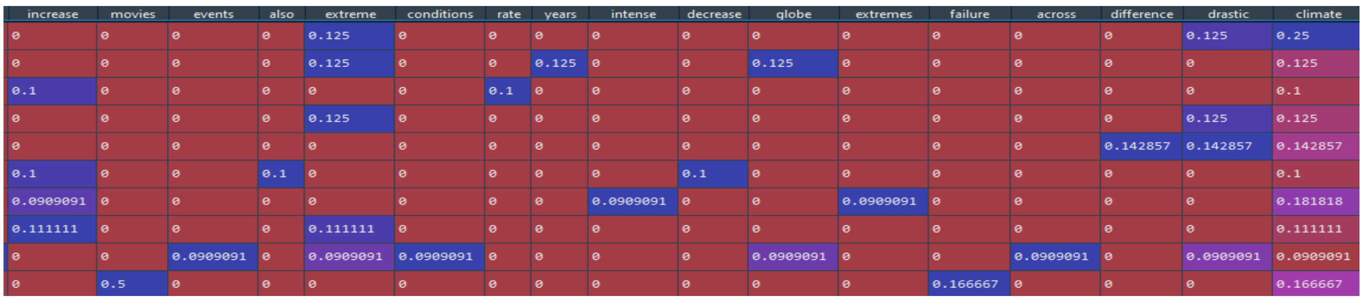


Term Frequence (TF) is defined by the number of times a term t appears in a document divided by the total number of words in that document. It defines how frequently a term occurs in a document. Sometimes the length of some documents is longer, that would mean some terms appear more frequently than others compared to shorter documents. Hence, we divide it by the number of words / length of document. Since the examples I have used have a varied document length, I have normalized the data by dividing by length. There is one issue with calculating tf. It is that occurrence of stop words is very dominant. Hence, removing of stop words would give clearer result. Another way to deal with it is by calculating the idf which is specifies the rarity of the word.

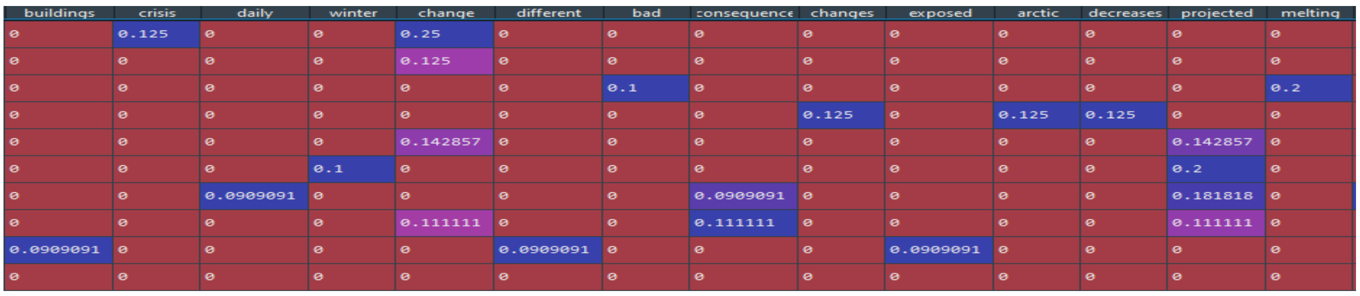
(I've made different word clouds for each document so that content of each document is clearly visible. Tried making one word cloud but it was too crowded and not clearly conveyed what I wished to)

In the doc 10 word cloud, you can observe the words movies, climate an failure are shown. That is because the frequency of these words is very high in that document. Score for movie=0.5 and for climate and failure is same i.e 1.6 (we can observe same font size for both these words)

In doc 3, melting word is used very frquently, hence the font size is larger as we are providing the tf score to word cloud. (score for melting=0.2)

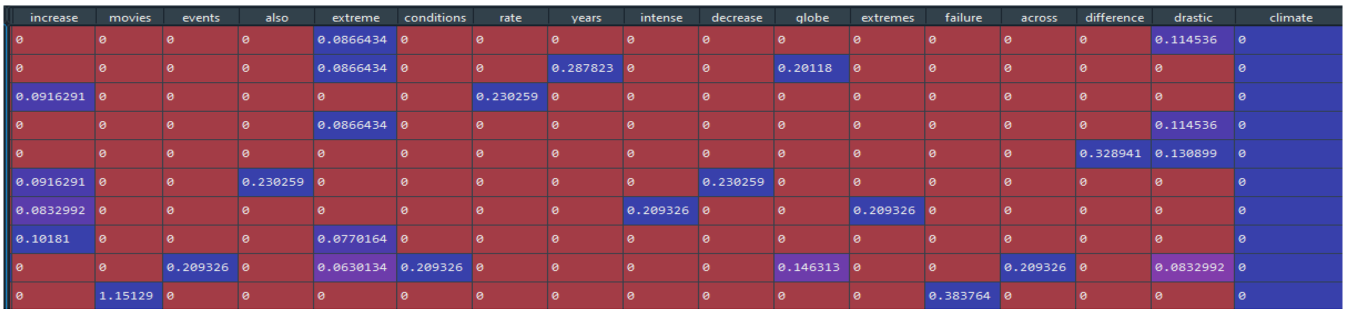


increase	movies	events	also	extreme	conditions	rate	years	intense	decrease	globe	extremes	failure	across	difference	drastic	climate
0	0	0	0	0.125	0	0	0	0	0	0	0	0	0	0	0.125	0.25
0	0	0	0	0.125	0	0	0.125	0	0	0.125	0	0	0	0	0	0.125
0.1	0	0	0	0	0	0.1	0	0	0	0	0	0	0	0	0	0.1
0	0	0	0	0.125	0	0	0	0	0	0	0	0	0	0	0.125	0.125
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.142857	0.142857	0.142857
0.1	0	0	0.1	0	0	0	0	0	0.1	0	0	0	0	0	0	0.1
0.0909091	0	0	0	0	0	0	0	0.0909091	0	0	0.0909091	0	0	0	0	0.181818
0.111111	0	0	0	0.111111	0	0	0	0	0	0	0	0	0	0	0	0.111111
0	0	0.0909091	0	0.0909091	0.0909091	0	0	0	0	0.0909091	0	0	0.0909091	0	0.0909091	0.0909091
0	0.5	0	0	0	0	0	0	0	0	0	0	0.166667	0	0	0	0.166667

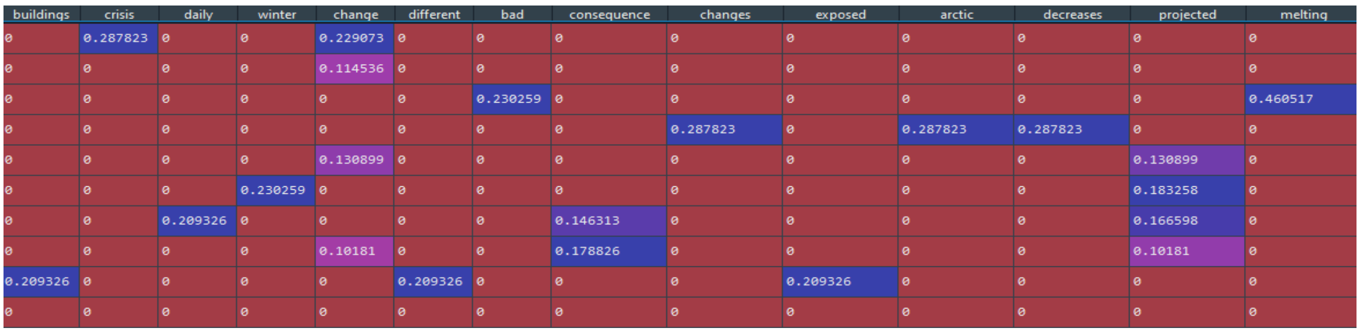


buildings	crisis	daily	winter	change	different	bad	consequence	changes	exposed	arctic	decreases	projected	melting
0	0.125	0	0	0.25	0	0	0	0	0	0	0	0	0
0	0	0	0	0.125	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.1	0	0	0	0	0	0	0.2
0	0	0	0	0	0	0	0	0.125	0	0.125	0.125	0	0
0	0	0	0	0.142857	0	0	0	0	0	0	0	0.142857	0
0	0	0	0.1	0	0	0	0	0	0	0	0	0.2	0
0	0	0.0909091	0	0	0	0	0.0909091	0	0	0	0	0.181818	0
0	0	0	0	0.111111	0	0	0.111111	0	0	0	0	0.111111	0
0.0909091	0	0	0	0	0.0909091	0	0	0	0.0909091	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

C. TF-IDF : It stands for term frequency inverse document frequency. It is a measure to understand how important a term or word is to a document in a collection of documents/corpus. There may be a few terms which appear more frequently but do not have much importance. To weigh down the frequent terms and make the less frequent words more prominent we use inverse document frequency.



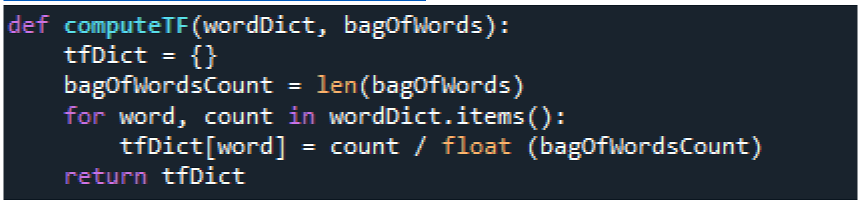
increase	movies	events	also	extreme	conditions	rate	years	intense	decrease	globe	extremes	failure	across	difference	drastic	climate
0	0	0	0	0.0866434	0	0	0	0	0	0	0	0	0	0	0.114536	0
0	0	0	0	0.0866434	0	0	0.287823	0	0	0.20118	0	0	0	0	0	0
0.0916291	0	0	0	0	0	0.230259	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0.0866434	0	0	0	0	0	0	0	0	0	0	0.114536	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.328941	0.130899	0
0.0916291	0	0	0.230259	0	0	0	0	0	0.230259	0	0	0	0	0	0	0
0.0832992	0	0	0	0	0	0	0	0.209326	0	0	0.209326	0	0	0	0	0
0.10181	0	0	0	0.0770164	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0.209326	0	0.0630134	0.209326	0	0	0	0	0.146313	0	0	0.209326	0	0.0832992	0
0	1.15129	0	0	0	0	0	0	0	0	0	0	0.383764	0	0	0	0



buildings	crisis	daily	winter	change	different	bad	consequence	changes	exposed	arctic	decreases	projected	melting
0	0.287823	0	0	0.229073	0	0	0	0	0	0	0	0	0
0	0	0	0	0.114536	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0.230259	0	0	0	0	0	0	0.460517
0	0	0	0	0	0	0	0	0.287823	0	0.287823	0.287823	0	0
0	0	0	0	0.130899	0	0	0	0	0	0	0	0.130899	0
0	0	0	0.230259	0	0	0	0	0	0	0	0	0.183258	0
0	0	0.209326	0	0	0	0	0.146313	0	0	0	0	0.166598	0
0	0	0	0	0.10181	0	0	0.178826	0	0	0	0	0.10181	0
0.209326	0	0	0	0	0.209326	0	0	0	0.209326	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0

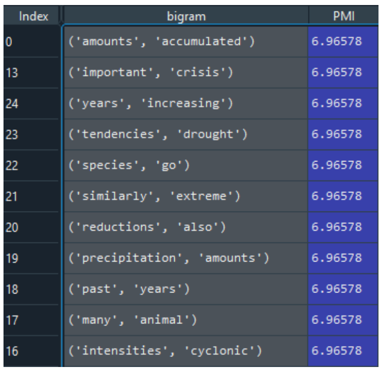
1. D. changes in relative ranking in term frequency matrix and tfidf matrix are as follows”
   - We can observe that in TF matrix, climate word has the maximum score/ranking 0.25. It is because that is the most frequent word. However, we can observe that in TF-IDF matrix, the score for the word climate becomes 0. It is because that word is present in all 10 documents and there the significance of that word is reduced. The is a potential drawback of TF-IDF since it is a fairly important word and it’s getting nullified weight after performing computations on it.
   - In TF matrix we can see the word building which is not really related with any other words in the corpus, it has a score of 0.09 which is very low as expected since it has low frequency. But the same word’s TF-IDF score is 0.20 which is fairly higher as per my expectations. I think this is observed because that term is very rare in the corpus and hence assigned a higher weight.

- One more such word is melting, it's TF score is 0.2 but the TF-IDF score is 0.4 which is comparatively higher. Again, it is so because of rarity of that word's occurrence in the corpus.
- Used the following code from ref: https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76



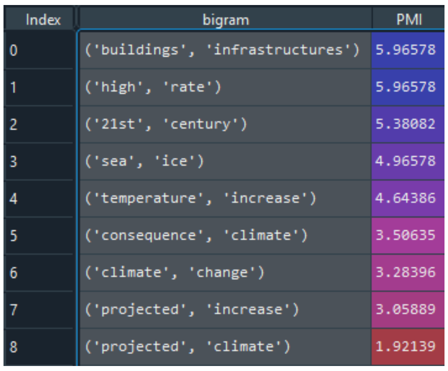
```python
def computeTF(wordDict, bagOfWords):
    tfDict = {}
    bagOfWordsCount = len(bagOfWords)
    for word, count in wordDict.items():
        tfDict[word] = count / float (bagOfWordsCount)
    return tfDict
```

2. PMI: Pointwise Mutual Information determines which words are more likely to occur together as opposed to occur independently. For example, the sentence "I like blue balloon" bigrams would be "I like", "like blue", "blue balloon" the first two does not make much sense but the last bigram implies the type of balloon. Therefore, PMI is important to make sense of the bigrams for further processing. We can code in python using following code
   bigramFinderA = nltk.collocations.BigramCollocationFinder.from_words(filtered_sentenceA)
   bigramPMITableA=pd.DataFrame(list(bigramFinderA.score_ngrams(bigrams.pmi)),columns=['bigram','PMI']).sort_values(by='PMI', ascending=False)
   Like in the following example, the top ranked words do not really make sense. For example, "years increasing", "amounts accumulated" etc are not very relevant terms



Index	bigram	PMI
0	('amounts', 'accumulated')	6.96578
13	('important', 'crisis')	6.96578
24	('years', 'increasing')	6.96578
23	('tendencies', 'drought')	6.96578
22	('species', 'go')	6.96578
21	('similarly', 'extreme')	6.96578
20	('reductions', 'also')	6.96578
19	('precipitation', 'amounts')	6.96578
18	('past', 'years')	6.96578
17	('many', 'animal')	6.96578
16	('intensities', 'cyclonic')	6.96578

Before minimum cut off

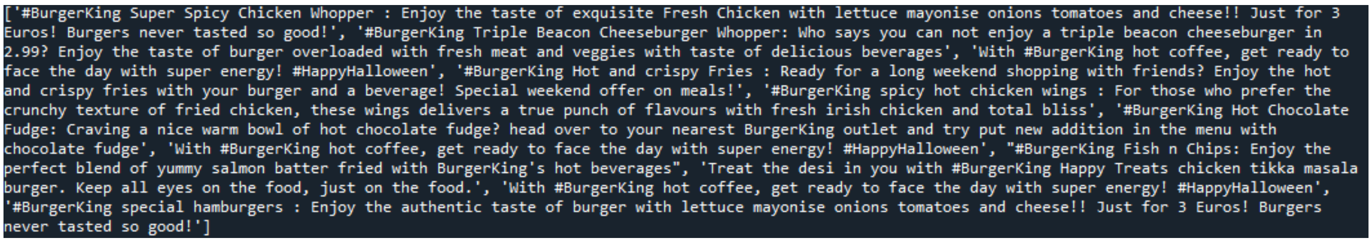


Index	bigram	PMI
0	('buildings', 'infrastructures')	5.96578
1	('high', 'rate')	5.96578
2	('21st', 'century')	5.38082
3	('sea', 'ice')	4.96578
4	('temperature', 'increase')	4.64386
5	('consequence', 'climate')	3.50635
6	('climate', 'change')	3.28396
7	('projected', 'increase')	3.05889
8	('projected', 'climate')	1.92139

After minimum cut off

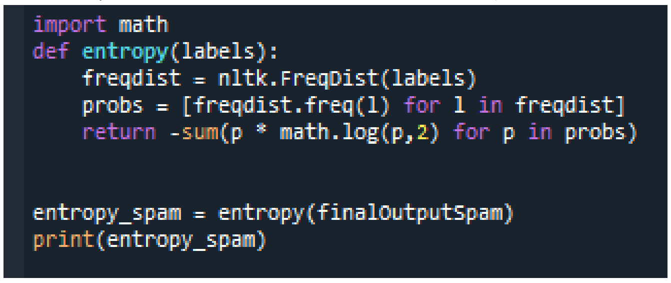
   We can apply filters bigramFinderA.apply_freq_filter(2) to introduce a minimum score. This will give better results. After applying filters, we get the following result. Here words like buildings and infrastructures, 21st century, high rate, sea ice etc are highly ranked and they are relevant and makes sense together. Issue with PMI is that it over estimates low frequency words. Possible solution may be using minimum frequency cut off. Nevertheless, PMI is very informative when it comes to understanding bigrams.

3. Entropy: is the measure of disorder. That means lower the entropy, there is more repetitiveness. High entropy means less repetitiveness and more disorder. It is defined as sum of the probability of each label times the log probability of the same label.
   - Used the following data for spam tweets



```
['#BurgerKing Super Spicy Chicken Whopper : Enjoy the taste of exquisite Fresh Chicken with lettuce mayonise onions tomatoes and cheese!! Just for 3
Euros! Burgers never tasted so good!', '#BurgerKing Triple Beacon Cheeseburger Whopper: Who says you can not enjoy a triple beacon cheeseburger in
2.99? Enjoy the taste of burger overloaded with fresh meat and veggies with taste of delicious beverages', 'With #BurgerKing hot coffee, get ready to
face the day with super energy! #HappyHalloween', '#BurgerKing Hot and crispy Fries : Ready for a long weekend shopping with friends? Enjoy the hot
and crispy fries with your burger and a beverage! Special weekend offer on meals!', '#BurgerKing spicy hot chicken wings : For those who prefer the
crunchy texture of fried chicken, these wings delivers a true punch of flavours with fresh irish chicken and total bliss', '#BurgerKing Hot Chocolate
Fudge: Craving a nice warm bowl of hot chocolate fudge? head over to your nearest BurgerKing outlet and try put new addition in the menu with
chocolate fudge', 'With #BurgerKing hot coffee, get ready to face the day with super energy! #HappyHalloween', "#BurgerKing Fish n Chips: Enjoy the
perfect blend of yummy salmon batter fried with BurgerKing's hot beverages", 'Treat the desi in you with #BurgerKing Happy Treats chicken tikka masala
burger. Keep all eyes on the food, just on the food.', 'With #BurgerKing hot coffee, get ready to face the day with super energy! #HappyHalloween',
'#BurgerKing special hamburgers : Enjoy the authentic taste of burger with lettuce mayonise onions tomatoes and cheese!! Just for 3 Euros! Burgers
never tasted so good!']
```

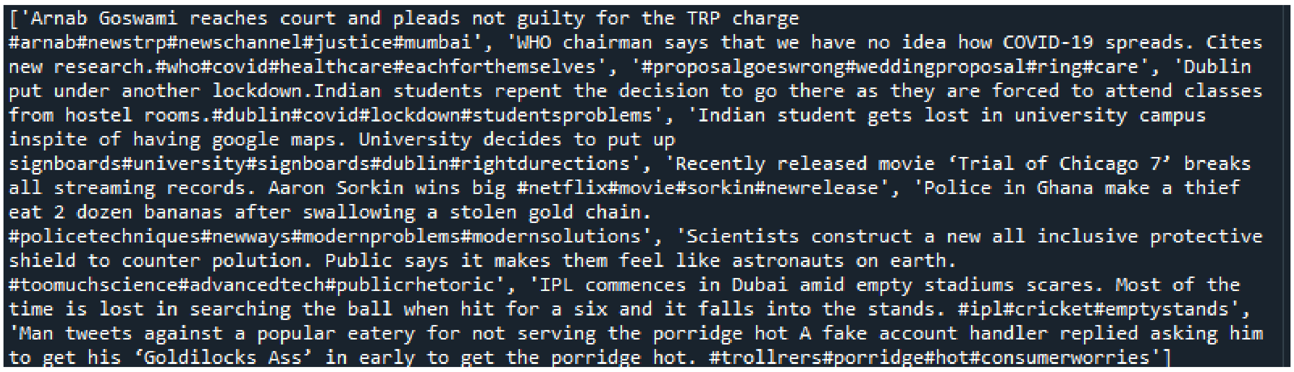
- Following code was used to calculate the entropy (Stackoverflow was used to solve coding doubts but used function provided in the tutorial for reference)



```
import math
def entropy(labels):
    freqdist = nltk.FreqDist(labels)
    probs = [freqdist.freq(l) for l in freqdist]
    return -sum(p * math.log(p,2) for p in probs)


entropy_spam = entropy(finalOutputSpam)
print(entropy_spam)
```

- Entropy calculated is : 6.276693352127533
- Used the following data for random tweets



```
['Arnab Goswami reaches court and pleads not guilty for the TRP charge
#arnab#newstrp#newschannel#justice#mumbai', 'WHO chairman says that we have no idea how COVID-19 spreads. Cites
new research.#who#covid#healthcare#eachforthemselves', '#proposalgoeswrong#weddingproposal#ring#care', 'Dublin
put under another lockdown.Indian students repent the decision to go there as they are forced to attend classes
from hostel rooms.#dublin#covid#lockdown#studentsproblems', 'Indian student gets lost in university campus
inspite of having google maps. University decides to put up
signboards#university#signboards#dublin#rightdurections', 'Recently released movie 'Trial of Chicago 7' breaks
all streaming records. Aaron Sorkin wins big #netflix#movie#sorkin#newrelease', 'Police in Ghana make a thief
eat 2 dozen bananas after swallowing a stolen gold chain.
#policetechniques#newways#modernproblems#modernsolutions', 'Scientists construct a new all inclusive protective
shield to counter polution. Public says it makes them feel like astronauts on earth.
#toomuchscience#advancedtech#publicrhetoric', 'IPL commences in Dubai amid empty stadiums scares. Most of the
time is lost in searching the ball when hit for a six and it falls into the stands. #ipl#cricket#emptystands',
'Man tweets against a popular eatery for not serving the porridge hot A fake account handler replied asking him
to get his 'Goldilocks Ass' in early to get the porridge hot. #trollrers#porridge#hot#consumerworries']
```

- Entropy calculated is:
  entropy_random = entropy(finalOutputRandom)
  print(entropy _random)
  7.0249934111911365
- Calculating the combined frequency, the result is as follows:
  entropy_cobined = entropy(finalOutputRandom +finalOutputSpam)
  print(entropy_cobined)
  7.543993306919609