Topic No. 12

Text Mining/Natural Language Processing Visualization

1. **Token frequency counts**
2. **Sentiment analysis**

    - Seaborn barplot

    - Visualizing n-grams

    - Treemap

    - Wordcloud

NLP visualizations for clear, immediate insights into text data

Exploratory Data Analysis for Natural Language Processing: A Complete Guide to Python Tools
- Shahul ES (July 20th, 2021)

Exploratory data analysis is one of the most important parts of any machine learning workflow and Natural Language Processing is no different. But which tools you should choose to explore and visualize text data efficiently?

https://neptune.ai/blog/exploratory-data-analysis-natural-language-processing-tools

Using Plotly Express and Dash to explore data and present outputs in natural language processing (NLP) projects.
- JP Hwang (March 30, 2020)

Extracting information from text remains a difficult, yet important challenge in the era of big data. Whether it comes to **customer feedback, social media posts, or the news**, the sheer volume of data to be analyzed can overwhelm information to be extracted.

https://medium.com/plotly/nlp-visualisations-for-clear-immediate-insights-into-text-data-and-outputs-9ebfab168d5b

This is where modern natural language processing (NLP) tools come in. They can

1. **capture prevailing moods about a particular topic or product (sentiment analysis),**
2. **identify key topics from texts (summarization/classification), or**
3. **amazingly even answer context-dependent questions (like Siri or Google Assistant).**

ABC news headlines @kaggle.com (Australian Broadcasting Corporation)

Data Explorer

62.73 MB

abcnews-date-text.csv

1195191
unique values

```
import numpy as np
import pandas as pd
import seaborn as sns

news= pd.read_csv('E:/Data/abcnews-date-text.csv',
                  nrows=10000)
```

publish_date	headline_text
20030219	aba decides against community broadcasting licence
20030219	act fire witnesses must be aware of defamation
20030219	a g calls for infrastructure protection summit
20030219	air nz staff in aust strike for pay rise
20030219	air nz strike to affect australian travellers
20030219	ambitious olsson wins triple jump
20030219	antic delighted with record breaking barca
20030219	aussie qualifier stosur wastes four memphis match
20030219	aust addresses un security council over iraq
20030219	australia is locked into war timetable opp
20030219	australia to contribute 10 million in aid to iraq
20030219	barca take record as robson celebrates birthday in
20030219	bathhouse plans move ahead
20030219	big hopes for launceston cycling championship
20030219	big plan to boost paroo water supplies
20030219	blizzard buries united states in bills
20030219	brigadier dismisses reports troops harassed in
20030219	british combat troops arriving daily in kuwait
20030219	bryant leads lakers to double overtime win
20030219	bushfire victims urged to see centrelink
20030219	businesses should prepare for terrorist attacks
20030219	calleri avenges final defeat to eliminate massu
20030219	call for ethanol blend fuel to go ahead
20030219	carews freak goal leaves roma in ruins
20030219	cemeteries miss out on funds
20030219	code of conduct toughens organ donation regulations
20030219	commonwealth bank cuts fixed home loan rates

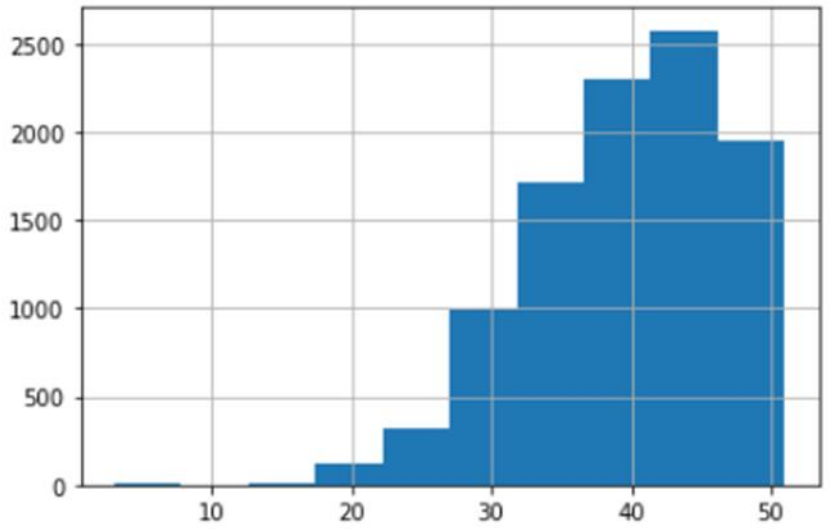
https://www.kaggle.com/therohk/million-headlines

Token frequency counts

```
# pip install wordcloud   - it needs to be installed.

import numpy as np
import pandas as pd
import seaborn as sns

# many rows (~28,000) of <id, sentence>
news= pd.read_csv('E:/abcnews-date-text.csv',nrows=10000)
print(news.head(10))

news['headline_text'].str.len().hist()
```

```
   publish_date                                      headline_text
0      20030219  aba decides against community broadcasting lic...
1      20030219     act fire witnesses must be aware of defamation
2      20030219     a g calls for infrastructure protection summit
3      20030219           air nz staff in aust strike for pay rise
4      20030219      air nz strike to affect australian travellers
5      20030219                  ambitious olsson wins triple jump
6      20030219         antic delighted with record breaking barca
7      20030219  aussie qualifier stosur wastes four memphis match
8      20030219       aust addresses un security council over iraq
9      20030219         australia is locked into war timetable opp
```

```
<AxesSubplot:>
```

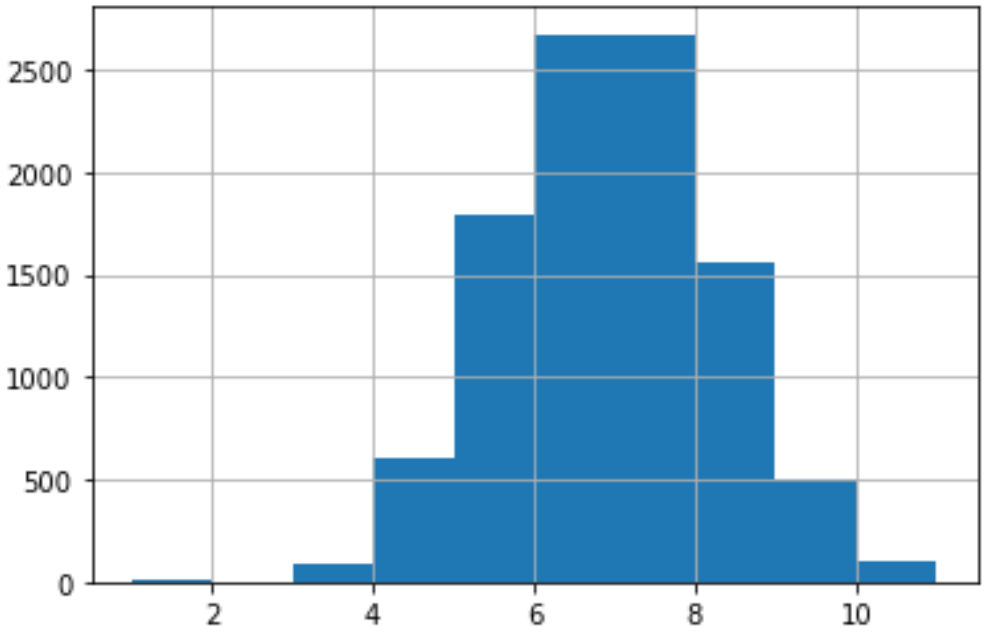



```
# This time, do histogram analysis for "mean" of word lengths.
news['headline_text'].str.split().apply(lambda x : [len(i) for i in x]).map(lambda x: np.mean(x)).hist()
```

Token Frequency Histogram Analysis

```
# pip install wordcloud   - it needs to be installed.

import numpy as np
import pandas as pd
import seaborn as sns

# many rows (~28,000) of <id, sentence>
news= pd.read_csv('E:/abcnews-date-text.csv',nrows=10000)
print(news.head(10))

news['headline_text'].str.len().hist()
```

	publish_date	headline_text
0	20030219	aba decides against community broadcasting lic...
1	20030219	act fire witnesses must be aware of defamation
2	20030219	a g calls for infrastructure protection summit
3	20030219	air nz staff in aust strike for pay rise
4	20030219	air nz strike to affect australian travellers
5	20030219	ambitious olsson wins triple jump
6	20030219	antic delighted with record breaking barca
7	20030219	aussie qualifier stosur wastes four memphis match
8	20030219	aust addresses un security council over iraq
9	20030219	australia is locked into war timetable opp



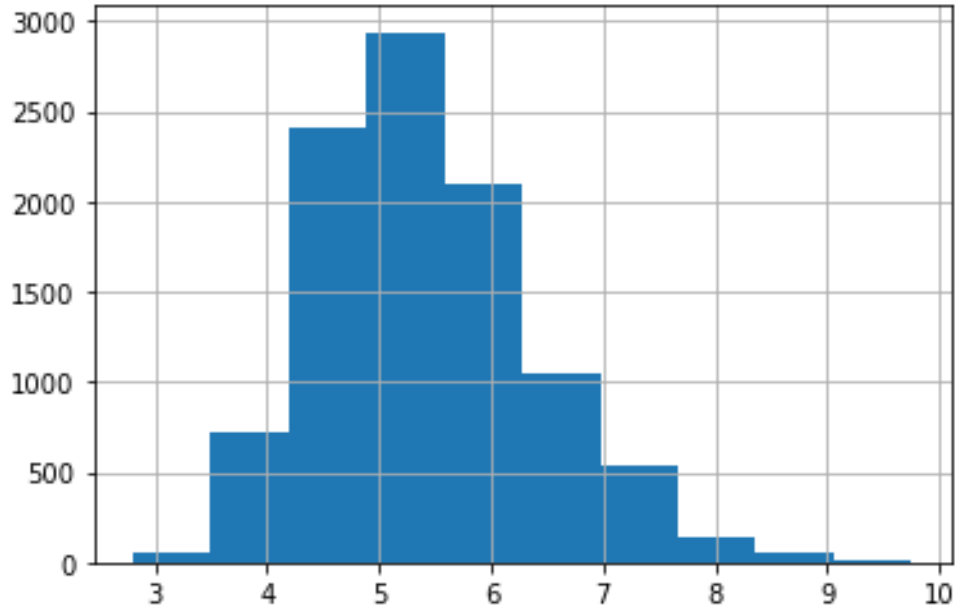

```
# This time, do histogram analysis after tokenized.
text = news['headline_text'] # create the unary table
# text[0].split()

text.str.split().map(lambda x: len(x)).hist()
```

```
# This time, do histogram analysis for "mean" of word lengths.
news['headline_text'].str.split().apply(lambda x : [len(i) for i in x]).map(lambda x: np.mean(x)).hist()
```

Mean Word Length Analysis

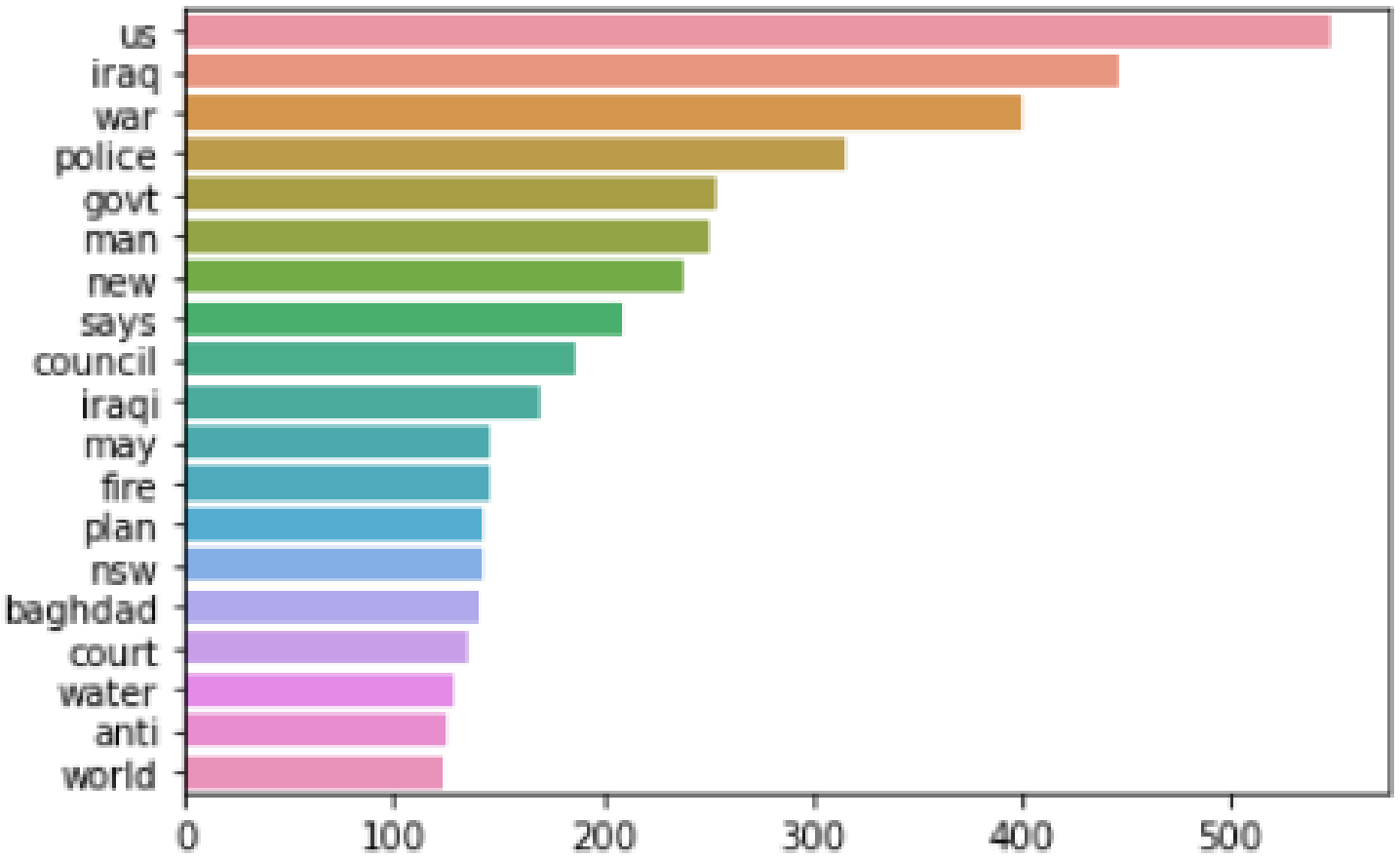
```
# This time, do histogram analysis for “mean” of word lengths.
news['headline_text'].str.split().apply(lambda x : [len(i) for i in x]).map(lambda x: np.mean(x)).hist()
```

	publish_date	headline_text
0	20030219	aba decides against community broadcasting lic...
1	20030219	act fire witnesses must be aware of defamation
2	20030219	a g calls for infrastructure protection summit
3	20030219	air nz staff in aust strike for pay rise
4	20030219	air nz strike to affect australian travellers
5	20030219	ambitious olsson wins triple jump
6	20030219	antic delighted with record breaking barca
7	20030219	aussie qualifier stosur wastes four memphis match
8	20030219	aust addresses un security council over iraq
9	20030219	australia is locked into war timetable opp

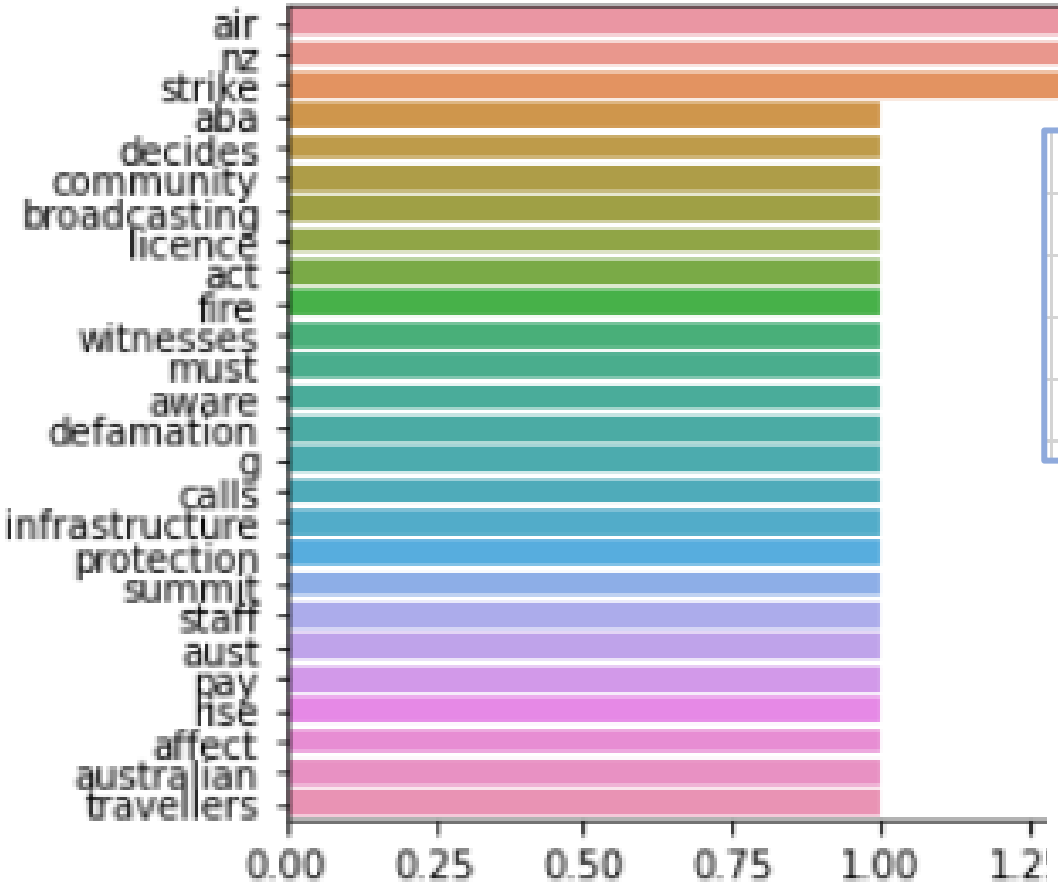


Text Mining Visualization: Seaborn Barplot

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Build stopwords
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
stop=set(stopwords.words('english'))

news= pd.read_csv('E:/Data/abcnews-date-
text.csv',nrows=10000)
print(news.head(10))

# Build the list of words "corpus"
corpus=[]
new= news['headline_text'].str.split()
new=new.values.tolist()
corpus=[word for i in new for word in i]

from collections import defaultdict
dic=defaultdict(int)
for word in corpus:
    if word in stop:
        dic[word]+=1

from collections import Counter
counter=Counter(corpus)
most=counter.most_common()

x, y= [], []
for word,count in most[:40]:
    if (word not in stop):
        x.append(word)
        y.append(count)

sns.barplot(x=y,y=x)
plt.show()
```

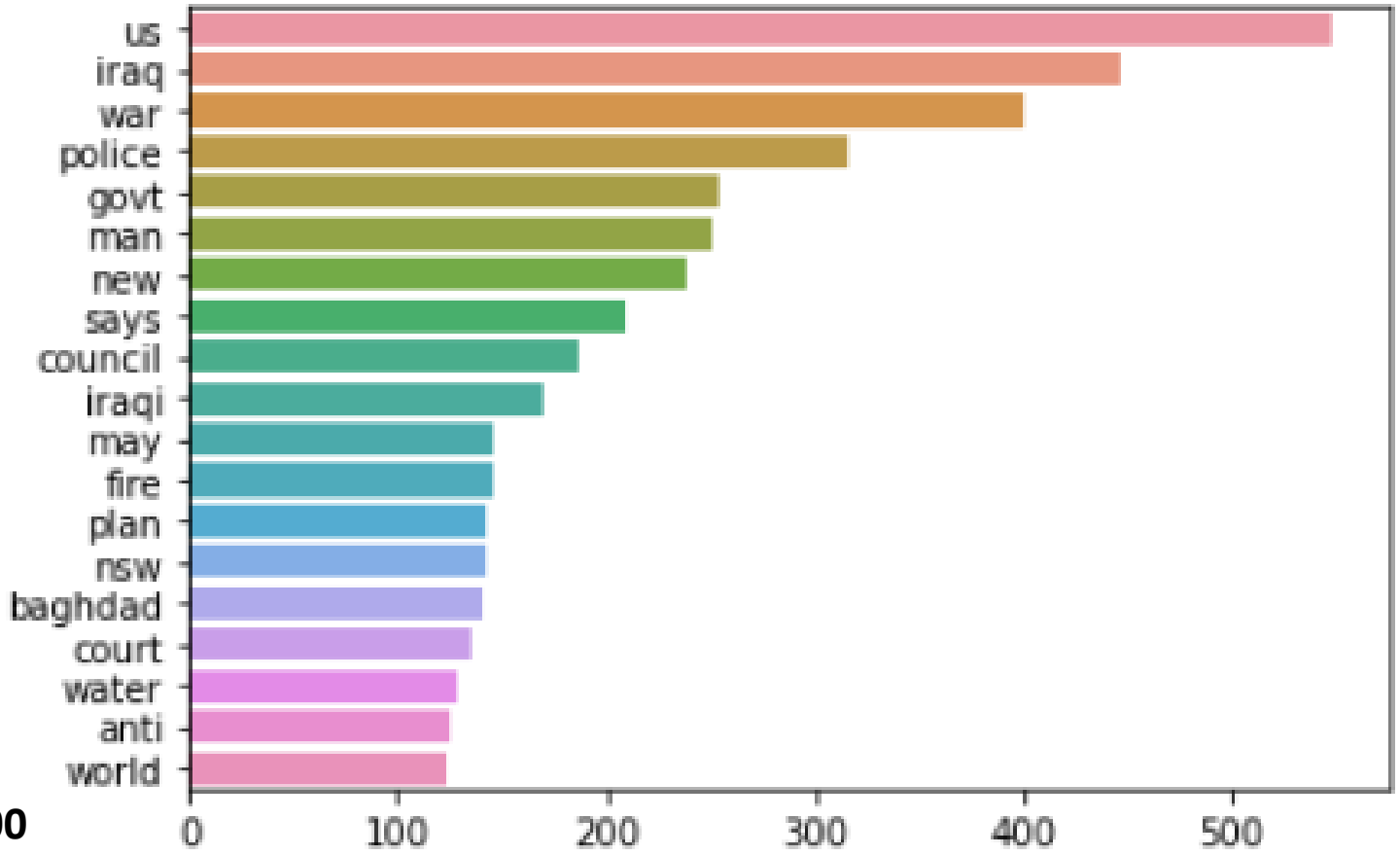


Volume needed



nrows = 50

aba decides against community broadcasting licence
act fire witnesses must be aware of defamation
a g calls for infrastructure protection summit
air nz staff in aust strike for pay rise
air nz strike to affect australian travellers

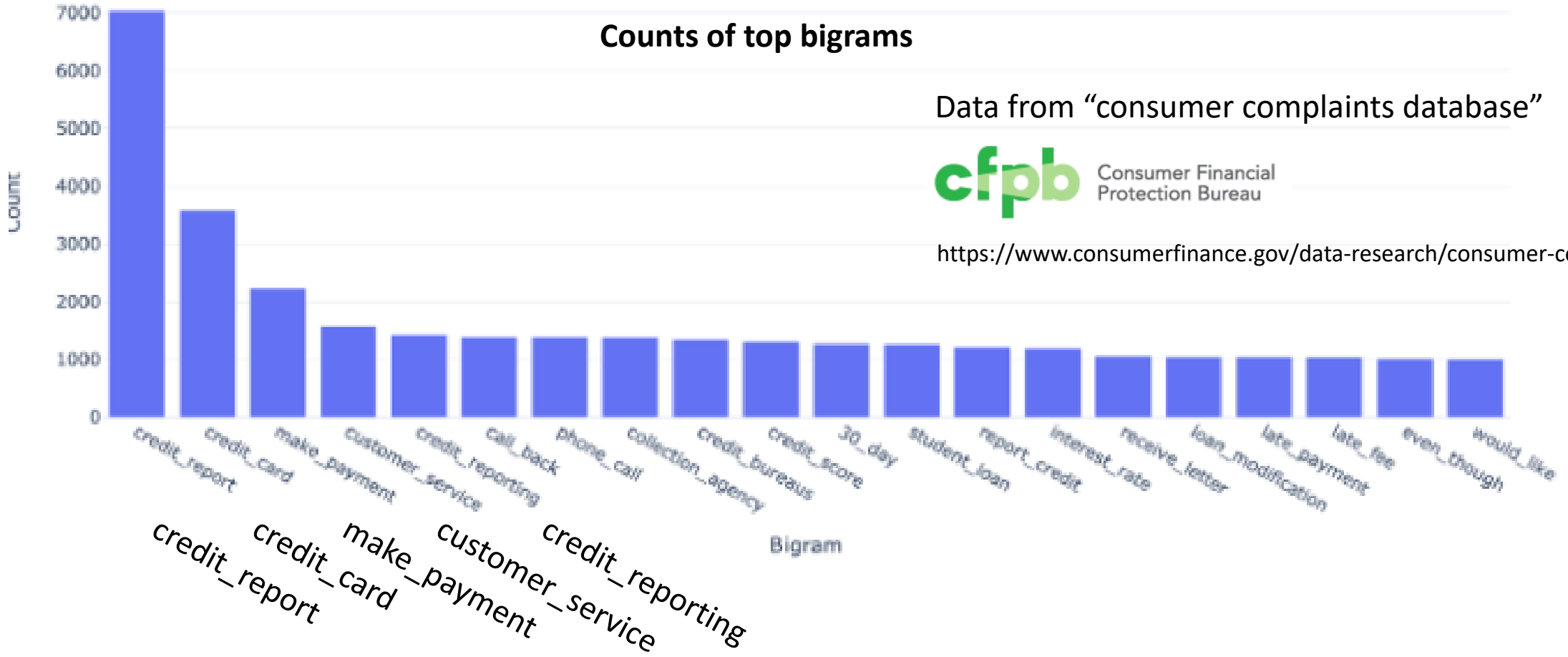


nrows = 10000

Visualizing n-grams

N-grams are simply sequences of tokens (words), and have many practical applications as well as being a great exploratory method. As **single words can only tell us so much**, let's move straight to plotting counts of top bigrams.

Bigram for complain types: Most of these bigrams appear to indicate sensible groups of complaint types, and the counts show the volume of each group (credit report and credit card related complaints appear to be most common).



Data from "consumer complaints database"

Consumer Financial Protection Bureau

https://www.consumerfinance.gov/data-research/consumer-complaints/

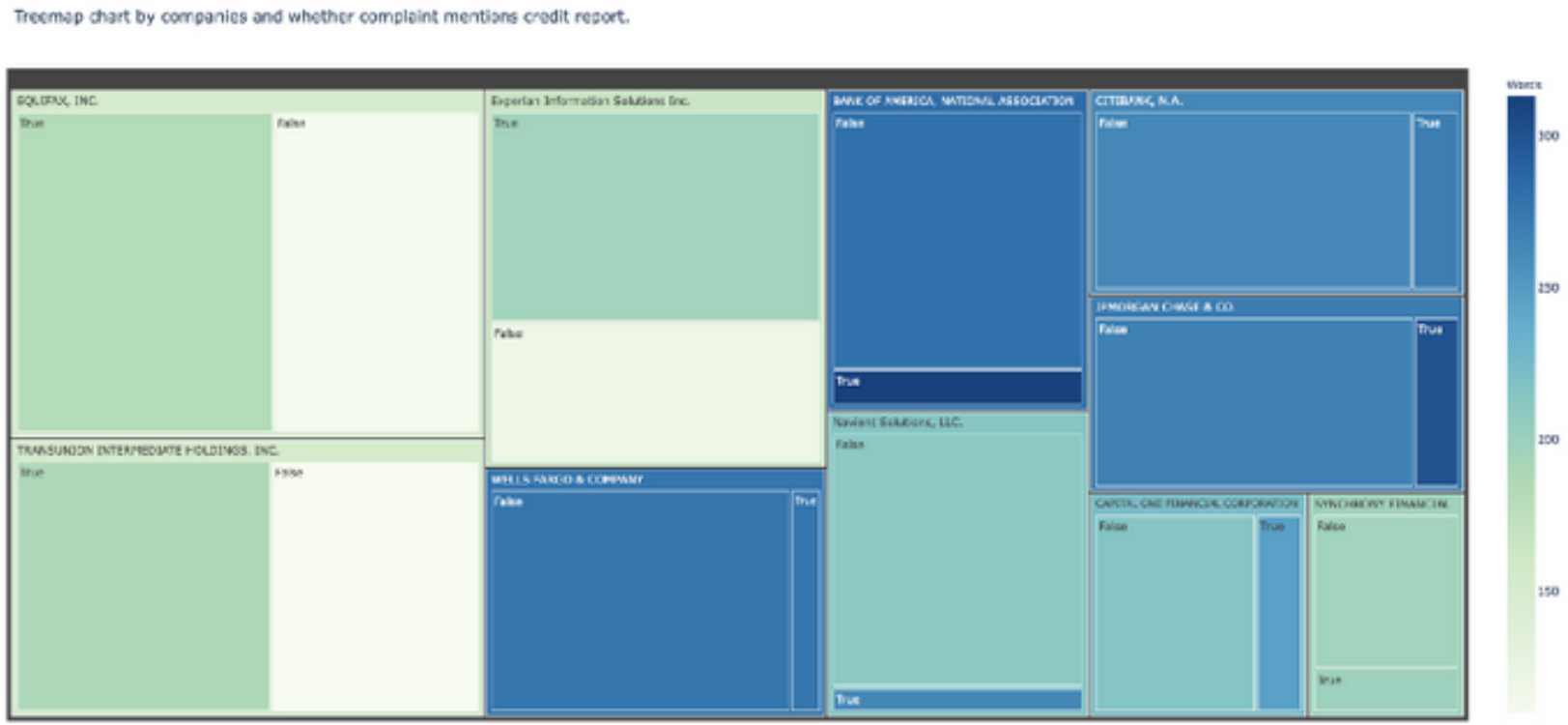
https://medium.com/plotly/nlp-visualisations-for-clear-immediate-insights-into-text-data-and-outputs-9ebfab168d5b

Treemap

To drill down further into this data, a **hierarchical visualization**, such as a treemap, could be used. This example below divides the data by company and then whether the phrase ‘credit report’ is included. Box sizes indicate group sizing, and color indicates average narrative length.

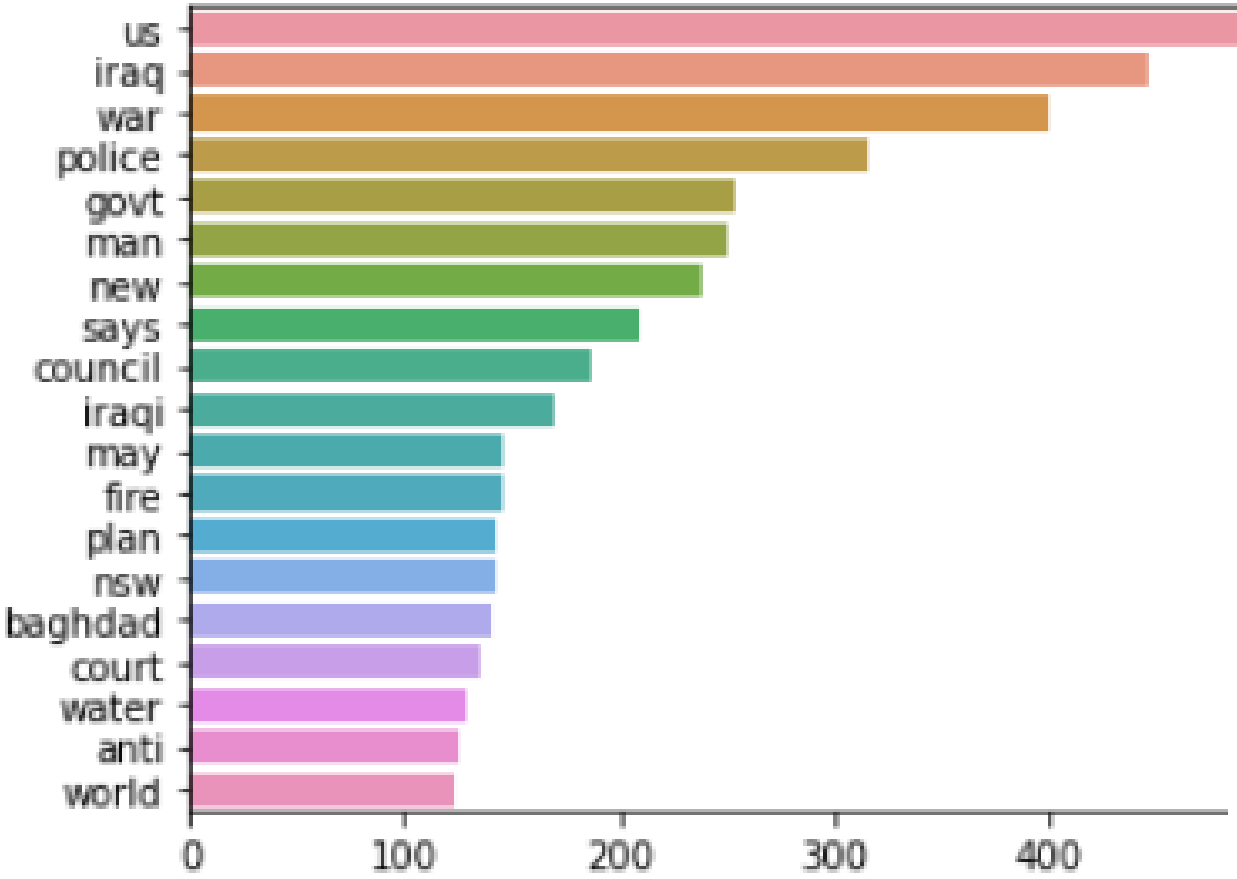
```
fig = px.treemap(top_comps_df, title='Treemap chart by companies and whether complaint mentions credit report.',
                 path=['Company', 'credit_report'], color='Words', color_continuous_scale=px.colors.sequential.GnBu)
fig.show()
```

Area = Weight



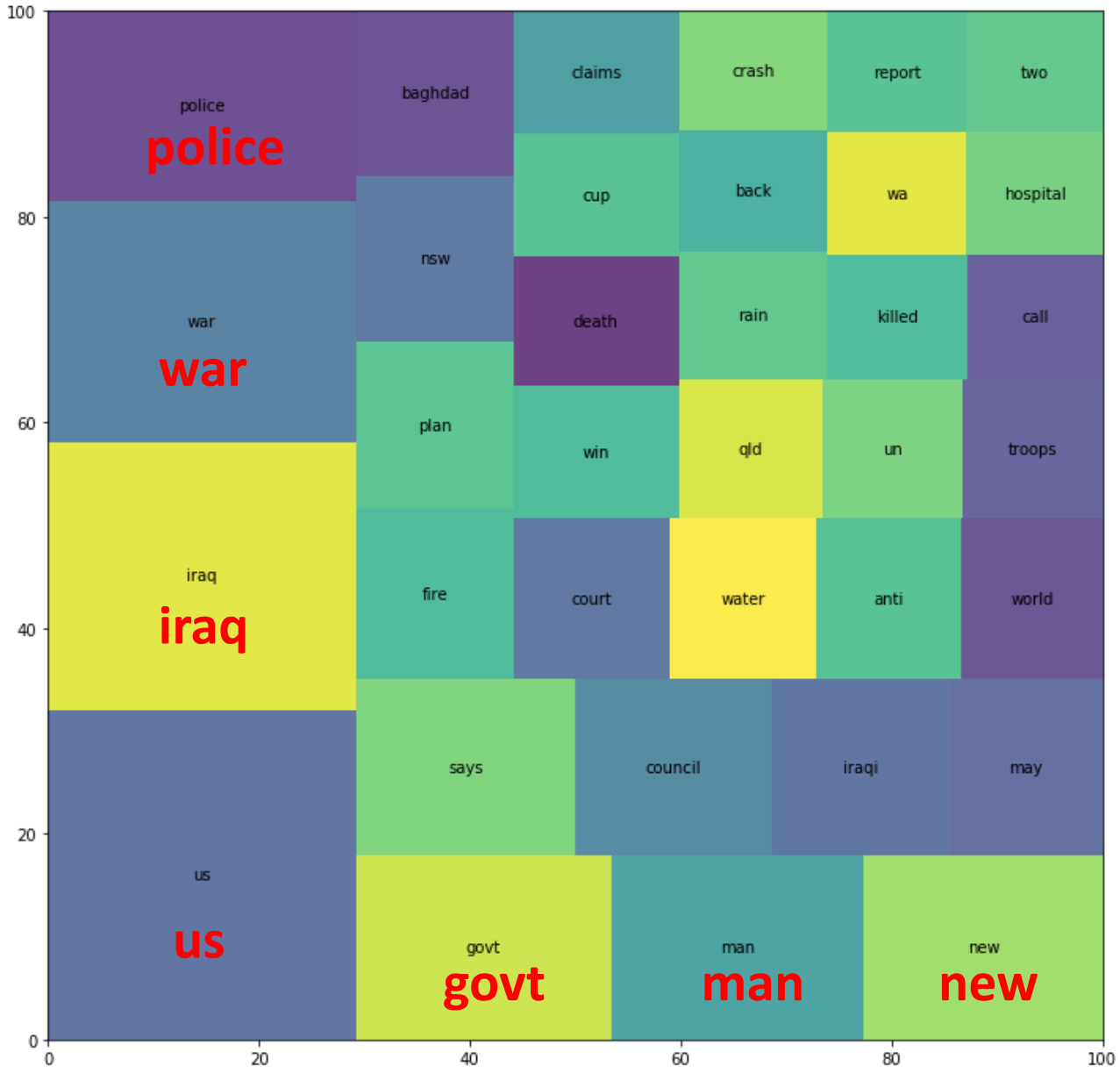
Treemap showing the total share of complaints, portion mentioning credit reports, and average lengths

https://medium.com/plotly/nlp-visualisations-for-clear-immediate-insights-into-text-data-and-outputs-9ebfab168d5b



Seaborn Barplot

Plotly Express Treemap

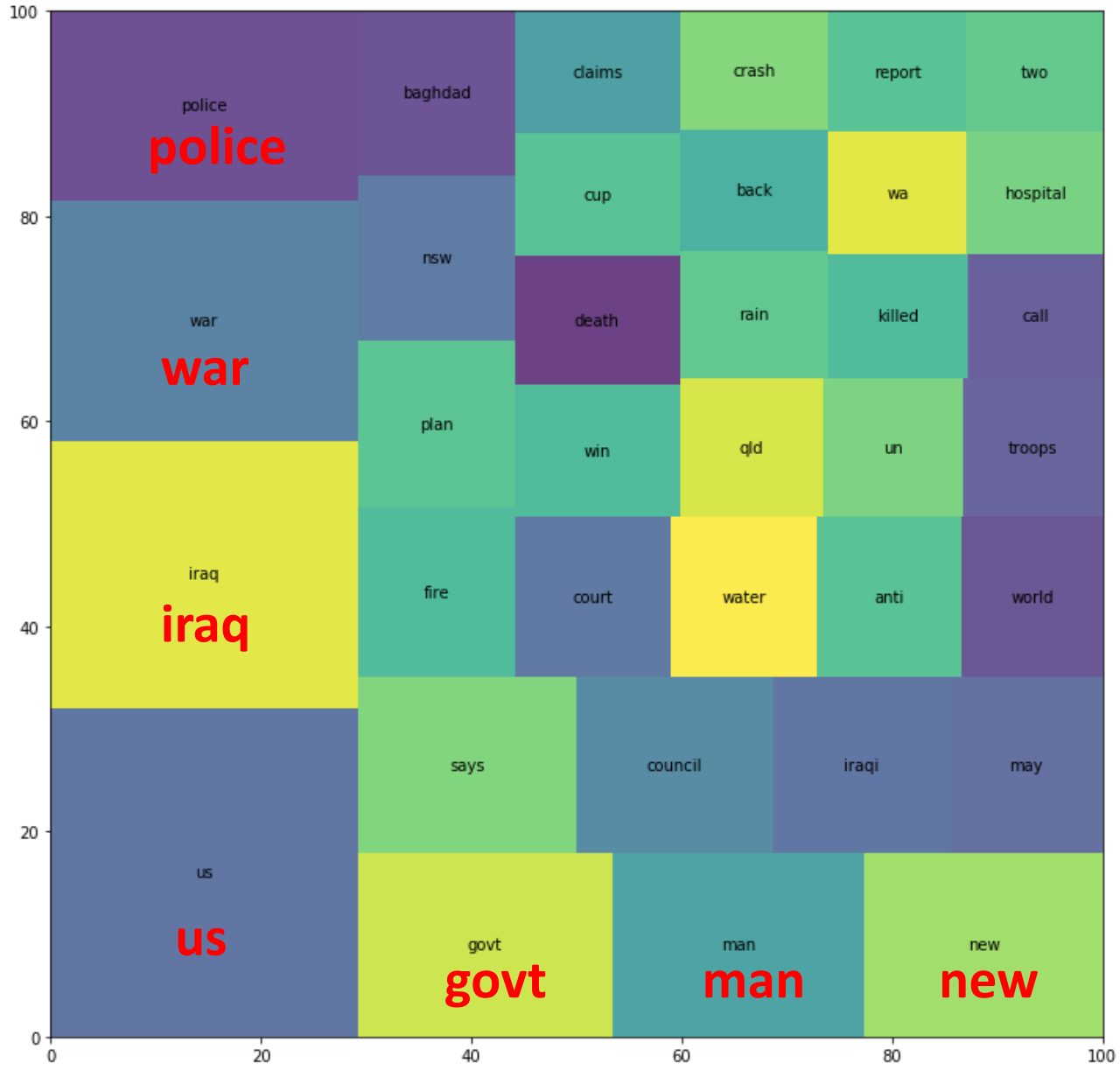


Analysis of ABC news headlines with nrows = 10000

Analysis of ABC news headlines with nrows = 10000

```
Topmost = []
token, cnt= [], []
for word,count in most[:60]:
    if (word not in stop):
        token.append(word)
        cnt.append(count)
        Topmost.append([word,count])

df = pd.DataFrame(data=Topmost,
columns=('Token', 'Frequency'))

fig, ax = plt.subplots(1, figsize =
(12,12))
squarify.plot(sizes=df2['Frequency'],
              label=df2['Token'],
              alpha=.8 )
```

Plotly Express Treemap



Wordcloud

```
from wordcloud import WordCloud, STOPWORDS
stopwords = set(STOPWORDS)

def show_wordcloud(data):
    wordcloud = WordCloud(
        background_color= 'white',
        stopwords=stopwords,
        max_words=50,
        max_font_size=30,
        scale=3,
        random_state=1)

    wordcloud=wordcloud.generate(str(data))

    fig = plt.figure(1, figsize=(12, 12))
    plt.axis('off')

    plt.imshow(wordcloud)
    plt.show()

show_wordcloud(corpus)
```

Why so many stop words?

Why are stop words not being excluded from the word cloud? → Set collocations=False

```
from wordcloud import WordCloud
from matplotlib import pyplot as plt
text = "The bear sat with the cat. They were good friends. " + \
       "My friend is a bit bear like. He's lovely. " + \
       "The bear, the cat, the dog and me were all sat " + \
       "there enjoying the view. You should have seen it. " + \
       "The view was absolutely lovely. " + \
       "It was such a lovely day. The bear was loving it too."

cloud = WordCloud(collocations=False, background_color='white', max_words=10).generate(text)
plt.imshow(cloud, interpolation='bilinear')
plt.axis('off')
plt.show()
```

collocations=True

collocations=False

Why are stop words not being excluded from the word cloud? → Set collocations=False

The default for a Wordcloud is that collocations=True, so frequent phrases of **two adjacent words** are included in the cloud - and importantly for your issue, with collocations the removal of stopwords is different, so that for example **“Thank you”** is a valid collocation and may appear in the generated cloud even though “you” is in the default stopwords. Collocations which contain only stopwords are removed.

```
text =
"The bear sat with the cat. They were good friends. " + \
"My friend is a bit bear like. He's lovely. " + \
"The bear, the cat, the dog and me were all sat " + \
"there enjoying the view. You should have seen it. " + \
"The view was absolutely lovely. " + \
"It was such a lovely day. The bear was loving it too."
```

Why “The bear” is broken?

collocations=True

collocations=False

https://stackoverflow.com/questions/61953788/why-are-stop-words-not-being-excluded-from-the-word-cloud-when-using-pythons-wo

Word Clouds: Unfortunately, the Status Quo in CX Platforms

Why word clouds harm insights

I still remember the first time I encountered a word cloud and was distinctly underwhelmed…

With my degree in Linguistics, I shuddered at this butchering of a single concept into three words. But a crude approach to dealing with language is only one of the reasons why word clouds aren't a viable form of visualizing data.

Alyona Medelyan PhD

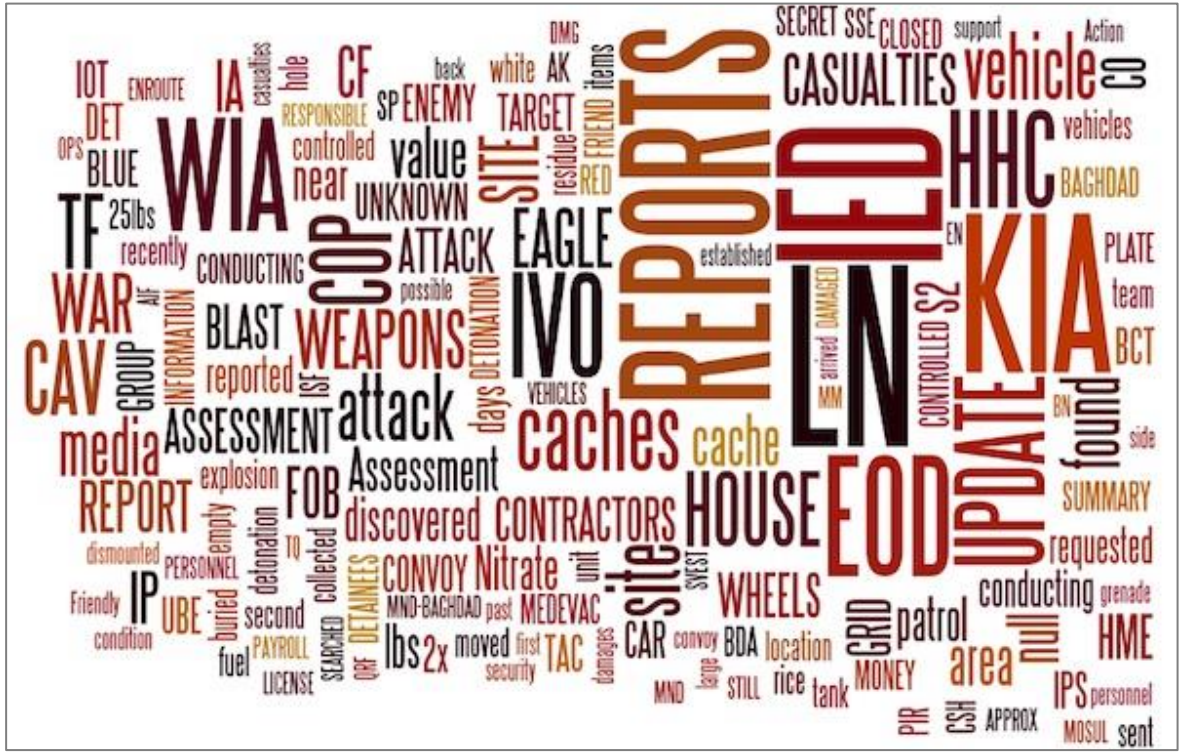
CEO and Co-Founder

https://getthematic.com/insights/word-clouds-harm-insights/

Word Clouds: the Mullets of Data Storytelling

Data expert Jacob Harris believes that visualizations are a form of storytelling. A good story does not overwhelm you with unnecessary information. A good story provides context to help you understand the subject. A good story leads you to the right conclusions.

According to Harris, word clouds "throw all of these principles out of the window", lead to the wrong conclusions about the data and are therefore harmful. As an example he shows these two visualizations derived from the same datasets:



Word Clouds: Five Major Shortcomings

1. Word clouds do not capture words that mean the same thing.

2. Word clouds do not capture complex themes.

3. Word clouds lack context.

4. Word clouds are prone to bias

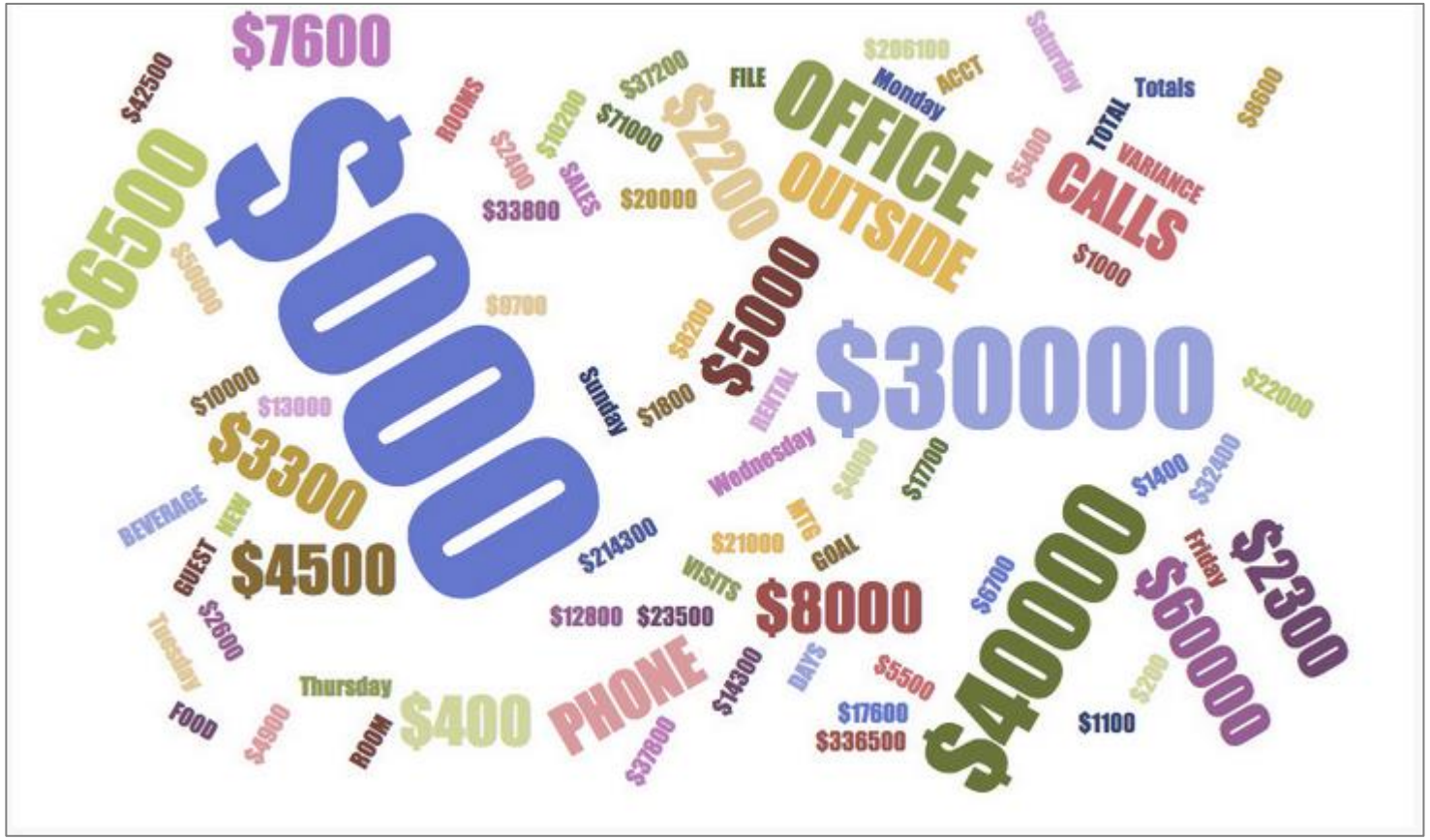
5. Word clouds obscure the relative importance of themes.

Alyona Medelyan PhD

CEO and Co-Founder

So, before you use a word cloud again in a report, think about this:

Would you take hard numbers like sale amounts for each week of the year, multiply each by a random amount between 1 and 5, delete some of them and then display the final numbers jumbled as a cloud?



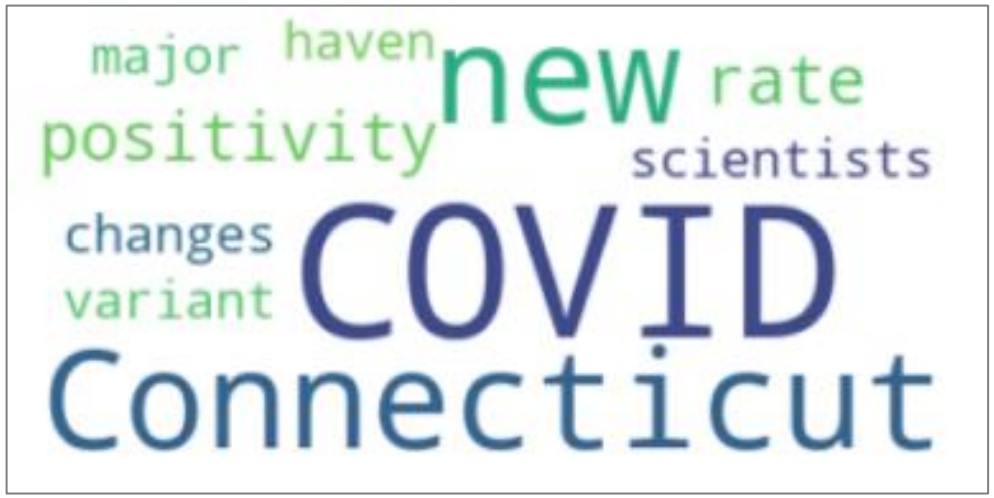
https://getthematic.com/insights/word-clouds-harm-insights/

Word Clouds: May not be that bad? Or is it still bad?

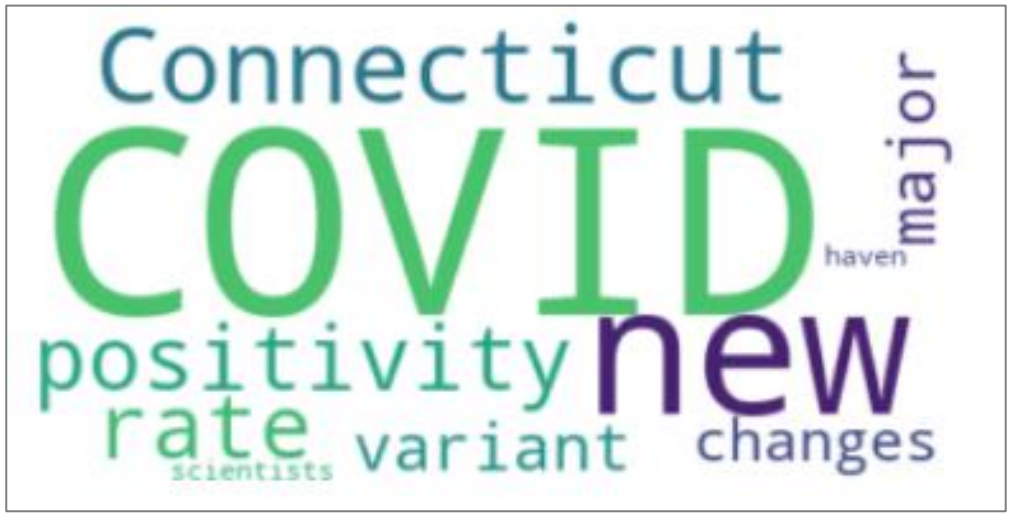
Using 6 Internet news headlines in Nov 17, 2021:

This new COVID-19 variant has major changes that scientists haven't seen before.
8 more die in New Hampshire of COVID-19 as hospitalizations rise.
The availability of new COVID-19 treatments will help low-income countries.
Connecticut records highest single-day positivity rate since August as COVID-19 numbers continue to increase.
COVID-19 outbreak at Connecticut nursing home kills 8, infects 89.
As Connecticut COVID positivity rate hits two-month high, officials push boosters.

`collocations=True:`



`collocations=False`



https://stackoverflow.com/questions/61953788/why-are-stop-words-not-being-excluded-from-the-word-cloud-when-using-pythons-wo