

# Identify Top Topics using Word Cloud



Karan Bhanot

Jan 18 · 4 min read



Photo by AbsolutVision on Unsplash

I was recently working with textual data when I discovered Word Clouds. I was really fascinated by how they could reveal so much information just through an image and how easily they could be created through a library. Thus, I decided to work on a quick project to understand them.

**Word clouds or tag clouds** are graphical representations of word frequency that give greater prominence to words that appear more frequently in a source text. —  
*BetterEvaluation*

Basically, Word Clouds display a set of words in the form of a cloud. The more frequent a word appears in the text, the bigger it will become. Thus, by simply looking at the cloud, you can identify the big words and hence the top topics.

## Numerous Areas of Word Cloud Usage

I identified that word clouds can actually be used in many areas. Some of them are:

1. **Top topics on Social Media:** If we could read and get text of posts/tweets that users are sending out, we can extract the top words out of them and they could be used in the trending section to classify and organise posts/tweets under respective sections.
2. **Trending News Topics:** If we can analyse the text or headings of various news articles, we can extract the top words out of them and identify what are the most trending news topics around a city, country or the whole world.
3. **Navigation systems for Websites:** Whenever you visit a website that is driven by categories or tags, a word cloud can actually be created and the users can directly jump to any topic while knowing the relevance of the topic across the community.

## Project — Detecting top news topics

I worked on a project, where I took the dataset of news articles from here and created a word cloud from the headlines of the news articles. The complete code is present as a Jupyter notebook in the Word Cloud repository.

### Import libraries

While working with importing libraries, I identified that I did not have the package `wordcloud`. Jupyter provides an easy way to execute command line commands inside the notebook itself. Just use `!` before the command and it'll work like it is in a command line. I am using it to get the `wordcloud` package.

```
!pip install wordcloud
```

I now have all the libraries that I need so I import all of them.

```
1 import collections
2 import numpy as np
```

```
3 import pandas as pd
4 import matplotlib.cm as cm
5 import matplotlib.pyplot as plt
6 from matplotlib import rcParams
7 from wordcloud import WordCloud, STOPWORDS
8 %matplotlib inline
```

import\_wordcloud.py hosted with ❤ by GitHub

[view raw](#)

We get the libraries `numpy`, `pandas`, `matplotlib`, `collections` to use `Counter` and `wordcloud` to create our Word Cloud.

## Working with dataset

To begin with, I first import the dataset file into a pandas DataFrame. Note that the encoding of this file for proper reading is `latin-1`. Then, I output the column names to identify which one matches with the headings.

```
1 dataset = pd.read_csv('dataset.csv', encoding='latin-1')
2 dataset.columns
3 ## Output:
4 # Index(['author', 'date', 'headlines', 'read_more', 'text', 'ctext'], dtype='object')
```

import\_dataset.py hosted with ❤ by GitHub

[view raw](#)

We can see that there are 6 columns: `author`, `date`, `headlines`, `read_more`, `text` and `ctext`. However, in this project I will be working with headlines. So, I convert all the headlines to lower case using `lower()` method and combine them into a variable `all_headlines`.

```
1 all_headlines = ' '.join(dataset['headlines'].str.lower())
```

combine\_headings.py hosted with ❤ by GitHub

[view raw](#)

## Word Cloud

Now, we're ready to create our Word Cloud. After doing one round of analysis, I identified one of the top words being `will`. However, it does not provide any useful information on the topic. Thus, I included it in the set of stopwords so that it is not considered while identifying the top words from the headings.

```
1 stopwords = STOPWORDS
2 stopwords.add('will')
3
```

[view raw](#)

I then call the `WordCloud` method using these stopwords, keep the background of the output image as `white` and set maximum words to be `1000`. The image is saved as `wordcloud`.

```
1 rcParams['figure.figsize'] = 10, 20
2 plt.imshow(wordcloud)
3 plt.axis("off")
4 plt.show()
```

[view raw](#)

I use `rcParams` to define the size of the figure and set the `axis` as `off`. I then use `imshow` to display the image and `show` to show it.



## Word Cloud

From the image, we can clearly see the top two topics as `India` and `Delhi` . One can clearly see how useful a word cloud is to identify the top words in a collection of text.

We can even verify the top words using the bar charts.

```
1 filtered_words = [word for word in all_headlines.split() if word not in stopwords]
```

```
2 counted_words = collections.Counter(filtered_words)
3
4 words = []
5 counts = []
6 for letter, count in counted_words.most_common(10):
7     words.append(letter)
8     counts.append(count)
```

most\_common\_words.py hosted with ❤ by GitHub

[view raw](#)

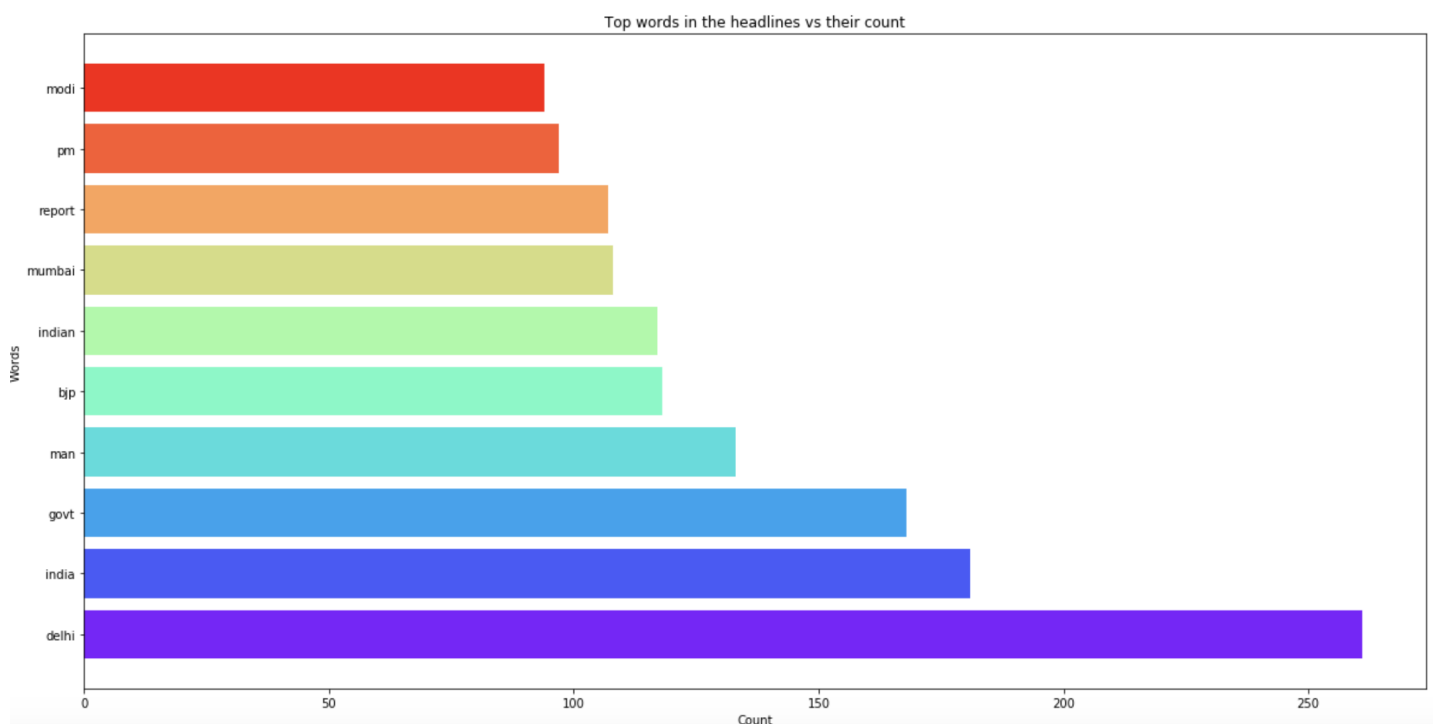
I first get `filtered_words` by splitting all words from the combined headings while avoiding the stopwords. Then, I used `Counter` to count the frequency of each word. I then extract the top 10 words and their count.

```
1 colors = cm.rainbow(np.linspace(0, 1, 10))
2 rcParams['figure.figsize'] = 20, 10
3
4 plt.title('Top words in the headlines vs their count')
5 plt.xlabel('Count')
6 plt.ylabel('Words')
7 plt.barh(words, counts, color=colors)
```

plot\_barchart.py hosted with ❤ by GitHub

[view raw](#)

Next, I plot the data and label the axis and define a title for the chart. I used `barh` to display a horizontal bar chart.



Bar Chart of top 10 most frequent words

This also is in alignment with the results from the Word Cloud. Moreover, as `Delhi` has a higher count, it is bolder and bigger than `India` in the Word Cloud.

## Conclusion

In this article, I discussed about what Word Clouds are, their potential application areas and a project that I worked on to understand them.

. . .

As always, please feel free to share your views and opinions.

[Data Science](#)   [Visualization](#)   [Analysis](#)   [Data](#)   [Data Visualization](#)

[About](#)   [Help](#)   [Legal](#)