

★ Get unlimited access to the best of Medium for less than \$1/week. [Become a member](#)



Photo by [Clément Falize](#) on [Unsplash](#)

How to Build an NLP Machine Learning App-End to End

NLP App using Streamlit and Python NLP Libraries



Senthil E · Following

Published in [Geek Culture](#)

8 min read · Apr 7, 2021

Listen

Share

More

Introduction:

Let's build a simple news article NLP app. I will walk through the steps with screenshots and code.

Contents:

1. *What is NLP?*
2. *NLP Applications.*
3. *Build the NLP App.*

What is NLP?

Natural Language Processing (NLP) is a branch of Artificial Intelligence (AI) that studies how machines understand human language. Its goal is to build systems that can make sense of text and perform tasks like translation, grammar checking, or topic classification.

NLP Applications:

NLP is one of the fastest-growing fields in machine learning. There is a lot of research going on and a lot of applications and used heavily by all the industries.

Some of the NLP applications are

- *Sentiment Analysis*
- *Chatbots*
- *Virtual Assistants/Voice Assistants*
- *Text Summarization*
- *Document Search*
- *Credit Scoring for Under-banked Clientele*
- *Text Extraction*
- *Speech Recognition*
- *Intent Classification*
- *AutoCorrect*
- *Email filtering*

- *Autocomplete*
- *Language Translation*

Build the NLP App:

The app is below



Video by the author — Uploaded the video in Youtube

About the App:

The app selects the RSS feeds from the top news publications. For example, I have selected the following news publications

- *NY Times*
- *LA Times*
- *Washington Post*
- *USA Today*
- *CNN*

If you want you can add your own news publications.

Then the NLP does the following

- *Intro*
- *Snapshot*
- *Unigrams*
- *Bigrams*
- *Trigrams*
- *WordCloud*
- *Text Stats*
- *Topic Modeling*
- *Entity Extraction*
- *Sentiment Analysis*
- *Text Summarization*
- *Parts of Speech*

Let's dive into the code now.

What is RSS Feeds?

According to Wikipedia

RSS (RDF Site Summary or Really Simple Syndication) is a web feed that allows users and applications to access updates to websites in a standardized, computer-readable format. These feeds can, for example, allow a user to keep track of many different websites in a single news aggregator. A standard XML file format ensures compatibility with many different machines/programs. RSS feeds also benefit users who want to receive timely updates from favourite websites or to aggregate data from many sites. Subscribing to a website RSS removes the need for the user to manually check the website for new content. Instead, their browser constantly monitors the site and informs the user of any updates. The browser can also be commanded to automatically download the new data for the user.

How to find the RSS Feeds?

For any browser like google chrome or Firefox follow the below steps

- Right-click on the website's page, and choose Page Source.
- In the new window that appears, use the “find” feature (Ctrl + F on a PC or Command + F on a Mac), and type in RSS. You'll find the feed's URL between the quotes after href=.

Check the below link for more details.

How to find RSS Link?

In our case the RSS link for NY Times is. We will consider NY Times as an example.

```
1 url_link = "https://rss.nytimes.com/services/xml/rss/nyt/US.xml"
```

np1 hosted with ❤ by GitHub

[view raw](#)

Image by the author

Based on the RSS link I will get the following information

- *Title*
- *Link*
- *Description*
- *Published*
- *Content*

Let's get the above information into a pandas data frame for easy processing.

The code to get the information in a pandas DF.

```

1  class RSSFeed():
2      feedurl = ""
3
4      global ndf
5
6      def __init__(self, paramrssurl):
7          print(paramrssurl)
8          self.feedurl = paramrssurl
9          self.parse()
10
11     def parse(self):
12         thefeed = feedparser.parse(self.feedurl)
13         global ndf
14         ndf = pd.DataFrame(columns=['title', 'link', 'description', 'published', 'content'])
15         published = thefeed.feed.get("description", "")
16         for thefeedentry in thefeed.entries:
17             title = thefeedentry.get("title", "")
18             link = thefeedentry.get("link", "")
19             descr = thefeedentry.get("description", "")
20             ndf = ndf.append({'title': title, 'link': link, 'description': descr, 'published': pu
21                             ignore_index=True})
22
23
24
25     df = pd.DataFrame(columns=['title', 'link', 'description', 'published', 'content'])
26     url_link = "https://rss.nytimes.com/services/xml/rss/nyt/US.xml"
27     RSSFeed(url_link)

```

np2 hosted with ❤ by GitHub

[view raw](#)

Image by the author

The dataframe consists of the following

	title	link	description	published	content
0	'I Was Failing': Bystanders Carry Guilt From W...	https://www.nytimes.com/2021/04/03/us/george-f...	Their pain has been mostly hidden for the last...	NaN	
1	Fed Up With Remote Learning, Governors Make a ...	https://www.nytimes.com/2021/04/03/us/covid-sc...	A bipartisan group of governors decided to fie...	NaN	
2	6 Takeaways From the First Week of the Derek C...	https://www.nytimes.com/2021/04/03/us/chauvin-...	It began with tears and shaky voices, and ende...	NaN	
3	Seven Infrastructure Problems in Urgent Need o...	https://www.nytimes.com/2021/04/02/us/infrastr...	The Biden administration has pledged a \$2 tril...	NaN	
4	Salmonella Outbreak Is Linked to Wild Birds. C...	https://www.nytimes.com/2021/04/04/health/bird...	The Centers for Disease Control and Prevention...	NaN	
5	Crash on California Border Highlights New Migr...	https://www.nytimes.com/2021/04/04/us/migrants...	In one of the deadliest border-related crashes...	NaN	
6	Amid Awakening, Asian-Americans Are Still Taki...	https://www.nytimes.com/2021/04/04/us/asians...	Divided by generation, ethnicity and class, bu...	NaN	
7	'Imminent Collapse' of Wastewater Reservoir in...	https://www.nytimes.com/2021/04/03/us/piney-po...	A potential deluge of nearly 400 million gallo...	NaN	
8	U.S. Tops Johnson & Johnson to Run Troubled Va...	https://www.nytimes.com/2021/04/03/us/politics...	The extraordinary move came just days after of...	NaN	
9	Noah Green, Capitol Suspect, Struggled Before...	https://www.nytimes.com/2021/04/03/us/politics...	Noah R. Green had been distressed but had no k...	NaN	
10	Fort Sill Soldiers Suspended in Sexual Assault...	https://www.nytimes.com/2021/04/03/us/fort-sil...	The U.S. Army Criminal Investigation Command i...	NaN	
11	April the Giraffe, Who Became an Internet Sens...	https://www.nytimes.com/2021/04/03/nyregion/ap...	April, 20, gained fame giving birth as million...	NaN	
12	Kemp Lashes M.L.B. as Republicans Defend Georg...	https://www.nytimes.com/2021/04/03/us/politics...	Gov. Brian Kemp of Georgia, who is starting hi...	NaN	
13	Forgotten Copy of Super Mario Bros. Sets Record...	https://www.nytimes.com/2021/04/03/business/su...	The game was bought as a Christmas gift in 198...	NaN	
14	Virus Variants Threaten to Draw Out the Pandem...	https://www.nytimes.com/2021/04/03/health/coro...	Declining infection rates overall masked a ris...	NaN	
15	How Trump Steered Supporters Into Unwitting Do...	https://www.nytimes.com/2021/04/03/us/politics...	Online donors were guided into weekly recruitin...	NaN	
16	Her Ballot Didn't Count. She Faces 5 Years in ...	https://www.nytimes.com/2021/04/03/us/texas-pr...	A Texas woman is appealing her conviction of v...	NaN	
17	The Latest Issue in Divorces: Who Gets the Emb...	https://www.nytimes.com/2021/04/03/health/liv...	Amid the pandemic, I.V.F. rates are on the ris...	NaN	
18	One Republican's Lonely Fight Against a Flood ...	https://www.nytimes.com/2021/04/03/us/politics...	After losing an ugly congressional race last y...	NaN	
19	Is Biden Missing His Chance on Guns?	https://www.nytimes.com/2021/04/03/us/politics...	President Biden spent decades pushing for gun...	NaN	

Image by the author

Now we have all the information except the content of the article. We can use the beautiful soup to get the content of the article.

```

1 #Beautiful Soup Code
2 @st.cache
3 def full_text(my_url):
4     import requests
5     from bs4 import BeautifulSoup
6     import pandas as pd
7     url = my_url
8     article = requests.get(url)
9     articles = BeautifulSoup(article.content, 'html.parser')
10    articles_body = articles.findAll('body')
11    p_blocks = articles_body[0].findAll('p')
12    p_blocks_df = pd.DataFrame(columns=['element_name', 'parent_hierarchy', 'element_text', 'element_text_Count'])
13    for i in range(0, len(p_blocks)):
14        parents_list = []
15        for parent in p_blocks[i].parents:
16            Parent_id = ''
17            try:
18                Parent_id = parent['id']
19            except:
20                pass
21        parents_list.append(parent.name + 'id: ' + Parent_id)
22        parent_element_list = ['' if (x == 'None' or x is None) else x for x in parents_list]
23        parent_element_list.reverse()
24        parent_hierarchy = ' -> '.join(parent_element_list)
25        p_blocks_df = p_blocks_df.append({"element_name": p_blocks[i].name
26                                         , "parent_hierarchy": parent_hierarchy
27                                         , "element_text": p_blocks[i].text
28                                         , "element_text_Count": len(str(p_blocks[i].text))
29                                         , ignore_index=True
30                                         , sort=False})
31    if len(p_blocks_df) > 0:
32        p_blocks_df_groupby_parent_hierarchy = p_blocks_df.groupby(by=['parent_hierarchy'])
33        p_blocks_df_groupby_parent_hierarchy_sum = p_blocks_df_groupby_parent_hierarchy[['element_text_Count']]
34        p_blocks_df_groupby_parent_hierarchy_sum.reset_index(inplace=True)
35        maxid = p_blocks_df_groupby_parent_hierarchy_sum.loc[
36            p_blocks_df_groupby_parent_hierarchy_sum['element_text_Count'].idxmax()
37            , 'parent_hierarchy']
38        merge_text = '\n'.join(p_blocks_df.loc[p_blocks_df['parent_hierarchy'] == maxid, 'element_text'])
39    return merge_text
40
41 text = full_text(url)

```

np3 hosted with ❤ by GitHub

[view raw](#)

Image by the author

Now we have all the information in a pandas dataframe

	title	link	description	published	content
0	'I Was Falling': Bystanders Carry Grief From W...	https://www.nytimes.com/2021/04/03/us/george-f...	Their pain has been mostly hidden for the last...	MINNEAPOLIS — Alyssa Funari needed a cord to c...	
1	Fed Up With Remote Learning, Governors Make a ...	https://www.nytimes.com/2021/04/03/us/covid-sc...	A bipartisan group of governors decided to file...	In Ohio, Gov. Mike DeWine offered school distr...	You will see that his respiration gets shall...
2	6 Takeaways From the First Week of the Derek C...	https://www.nytimes.com/2021/04/03/us/chauvin...	It began with tears and shaky voices, and ende...	Engineers say that when infrastructure works, ...	
3	Seven Infrastructure Problems in Urgent Need o...	https://www.nytimes.com/2021/04/02/us/infrastr...	The Biden administration has pledged a \$2 tril...	A salmonella outbreak linked to contact with w...	
4	Salmonella Outbreak Is Linked to Wild Birds, C...	https://www.nytimes.com/2021/04/04/health/bird...	The Centers for Disease Control and Prevention...	HOLTVILLE, Calif. — The maroon Ford Expedition...	
5	Crash on California Border Highlights New Migr...	https://www.nytimes.com/2021/04/04/us/migrants...	In one of the deadliest border-related crashes...	Divided by generation, ethnicity and class, bu...	When Mike Park first heard about the recent sh...
6	Amid Awakening, Asian-Americans Are Still Taki...	https://www.nytimes.com/2021/04/04/us/georgia...	A potential deluge of nearly 400 million gallo...	A reservoir in Florida that holds nearly 400 m...	
7	'Imminent' Collapse of Wastewater Reservoir in...	https://www.nytimes.com/2021/04/03/us/piney-po...	The extraordinary move came just days after of...	WASHINGTON — The Biden administration on Satur...	
8	U.S. Taps Johnson & Johnson to Run Troubled Va...	https://www.nytimes.com/2021/04/03/us/politics...	Noah R. Green had been distressed but had no k...	On the football field at Christopher Newport U...	
9	Noah Green, Capitol Suspect, Struggled Before ...	https://www.nytimes.com/2021/04/03/us/politics...			

Image by the author

Now we can start writing the code to do all the NLP functions

Please select the news org from the dropdown list

News

NY Times

Please select the Function

Functions

Intro

Snapshot

Unigrams

Bigrams

Trigrams

WordCloud

Text Stat

Traffic Matting

News Articles Analysis -NLP App

This app displays the news articles appeared in the top News Publications!




The New York Times (NYT or NY Times) is an American daily newspaper based in New York City with a worldwide readership. Founded in 1851, the Times has since won 130 Pulitzer Prizes (the most of any newspaper), and has long been regarded within the industry as a national "newspaper of record". It is ranked 18th in the world by circulation and 3rd in the U.S. The paper is owned by The New York Times Company, which is publicly traded. It has been governed by the Sulzberger family since 1896, through a dual-class share structure after its shares became publicly traded A

Image by the author — Screenshot of the NLP App

1. Intro:

Here I display the NY Times image and about NY Times information Wikipedia.

The code is below

```

1  if nlp == "Intro":
2      #st.write("this is intro")
3      image1 = Image.open(r'C:\Projects_Medium\New-York-Times-logo-500x281.jpg')
4      st.write( " ")
5      st.image(image1, width=300)
6      st.write(" ")
7      st.write(" ")
8      st.write(" ")
9      image = Image.open(r'C:\Projects_Medium\nytimes-building-ap-img.jpg')
10     st.image(image, width=700)
11     st.write(" ")
12     st.write(" ")
13     st.write(" ")
14     st.markdown(""""
15     The New York Times (NYT or NY Times) is an American daily newspaper based in New York
16     The paper is owned by The New York Times Company, which is publicly traded. It has been b
17     Since the mid-1970s, The New York Times has expanded its layout and organization, ad
18     The Times stayed with the broadsheet full-page set-up and an eight-column format for
19     """")

```

np4 hosted with ❤️ by GitHub

[view raw](#)

Image by the author

2. SnapShot:

The next item in the dropdown menu is the Snapshot. Here I display the dataframe and then the URL link.

#	title	link	description	publ
0	"I Was Falling": Bystander...	https://www.nytimes.com/2021/04/03/us/george-lloyd-derek-chauvin-trial.html	Their pain has been mostly...	
1	Fed Up With Remote Learning...	https://www.nytimes.com/2021/04/03/us/covid-schools-governors-reopening.html	It began with tears and s...	
2	6 Takeaways From the First...	https://www.nytimes.com/2021/04/03/us/chauvin-trial-takeaways.html	A bipartisan group of gov...	
3	Seven Infrastructure Projects...	https://www.nytimes.com/2021/04/03/us/infrastructure-projects-joe-biden.html	The Biden administration...	
4	Winfred Rembert, 75, Dies...	https://www.nytimes.com/2021/04/03/us/winfred-rembert-dead.html	Decades after nearly bei...	
5	Salmonella Outbreak Is Ll...	https://www.nytimes.com/2021/04/03/us/salmonella-outbreak-is-ll.html	The Centers for Disease C...	
6	Crash on California Border...	https://www.nytimes.com/2021/04/03/us/california-border-crash.html	In one of the deadliest b...	
7	Amid Awakening, Asian-Ameri...	https://www.nytimes.com/2021/04/03/us/asian-america-politics.html	Divided by generation, et...	
8	"Imminent" Collapse of Wu...	https://www.nytimes.com/2021/04/03/us/wuhan-collapse-of-wuhan.html	A potential deluge of ne...	
9	U.S. Tops Johnson & Johnson...	https://www.nytimes.com/2021/04/03/us/johnson-johnson.html	The extraordinary move ca...	
10	Noah Green, Capitol Suspe...	https://www.nytimes.com/2021/04/03/us/noah-green-capitol-suspect.html	Noah R. Green had been d...	

Image by the author — Screenshot of the NLP App

The code is below. Again just display the dataframe and the URL link.

```

1  if nlp == "Snapshot":
2      st.write(" ")
3      st.write(" ")
4      st.write(" ")
5      st.subheader('Display the dataframe')
6      st.write(" ")
7      st.write(" ")
8      st.write(" ")
9      st.dataframe(df)
10     st.write(" ")
11     st.write(" ")
12     st.write(" ")
13     st.markdown("""<style>
14         .big2-font {
15             font-size:30px !important;
16         }
17     </style>""", unsafe_allow_html=True)
18     st.markdown('<p class="big2-font">The no of articles :</p>', unsafe_allow_html=True)
19     st.write(df.shape[0])
20     st.write(" ")
21     st.write("The Url Link ")
22     for index, row in df.iterrows():
23         st.write(row['link'])

```

np5 hosted with ❤ by GitHub

[view raw](#)

Image by the author

3. Unigrams, Bigrams, and Trigrams:

Unigrams are single words, bigrams are two words, trigrams are three words, 4-grams are four words, 5-grams are five words, etc. N-gram is simply a sequence of N words.

Some of the applications where N-grams used in NLP

- *AutoComplete*
- *Auto Spell Check*
- *Check Grammar*
- *Virtual Assistant — Bots*

Want to learn more about N grams. Check the following youtube video

What is n-grams ?

The App looks like below for Trigrams

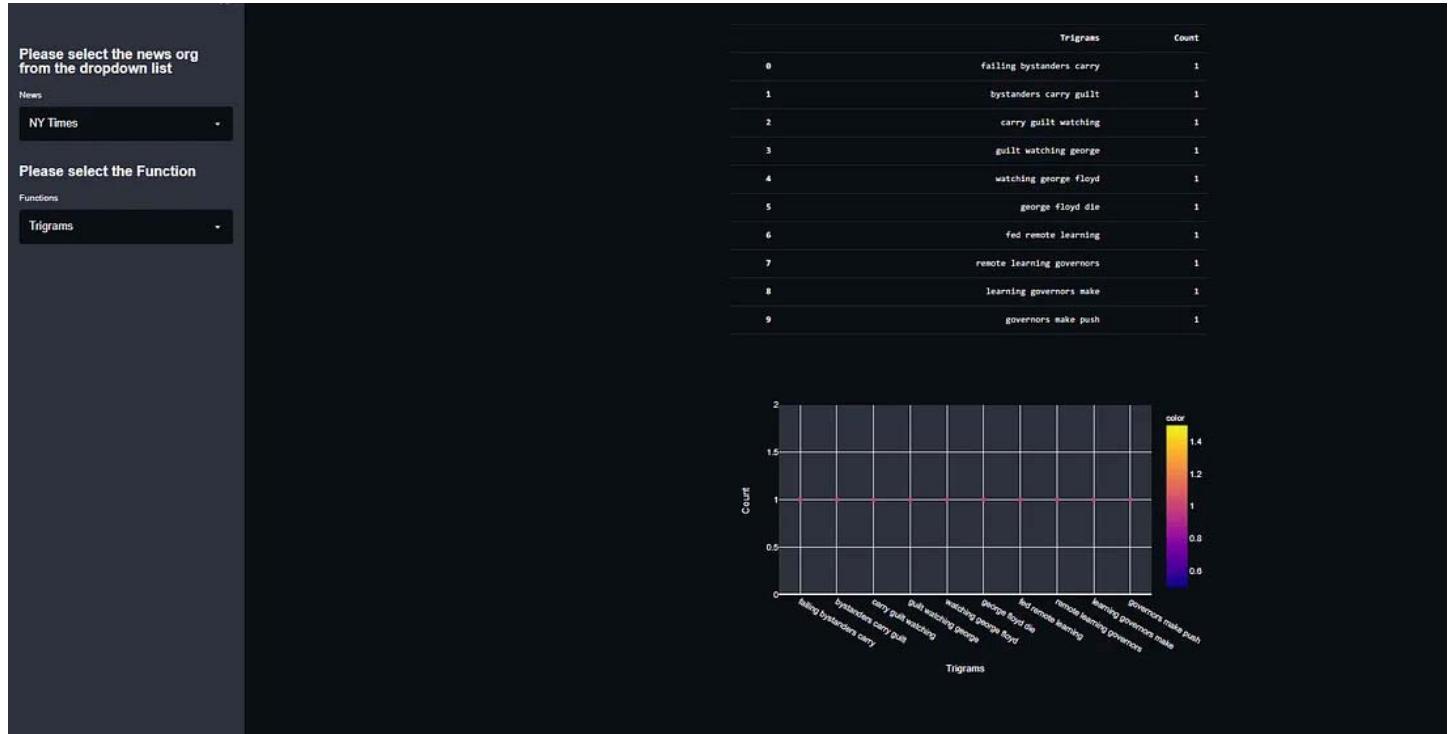


Image by the author — Screenshot of the NLP App

The code is

```

1  @st.cache
2  def get_top_n_bigram(corpus, n=None):
3      vec = CountVectorizer(ngram_range=(2, 2), stop_words='english').fit(corpus)
4      bag_of_words = vec.transform(corpus)
5      sum_words = bag_of_words.sum(axis=0)
6      words_freq = [(word, sum_words[0, idx]) for word, idx in vec.vocabulary_.items()]
7      words_freq = sorted(words_freq, key = lambda x: x[1], reverse=True)
8      return words_freq[:n]
9
10 if nlp == "Bigrams":
11     st.markdown("""<style>.big1-font {font-size:20px !important;}</style>
12 """, unsafe_allow_html=True)
13     st.write(' ')
14     st.markdown('<p class="big1-font">N Grams</p>', unsafe_allow_html=True)
15     st.write(' ')
16     st.markdown(
17         '<p class="big1-font">N-grams of texts are extensively used in text mining and n
18         unsafe_allow_html=True)
19     st.write(' ')
20     st.write(' ')
21     common_words = get_top_n_bigram(df_n['title'], 10)
22     df4 = pd.DataFrame(common_words, columns=['Bigrams', 'Count'])
23     st.table(df4)
24     fig = px.bar(df4, x='Bigrams', y='Count', color='Count', height=500)
25     st.plotly_chart(fig)

```

np6 hosted with ❤ by GitHub

[view raw](#)

Image by the author

4. WordCloud:

Google says a word cloud is “an image composed of words used in a particular text or subject, in which the size of each word indicates its frequency or importance.” So, the more often a specific word appears in your text, the bigger and bolder it appears in your word cloud. Wordcloud is used in a lot of applications like in the company website, try to extract from the customer feedback, employee feedback, etc.



Image by the author — Screenshot of the NLP App

The code is

```

1  if nlp == "WordCloud":
2      st.markdown("""<style>.big1-font {font-size:20px !important;}</style>
3      """, unsafe_allow_html=True)
4      st.write(' ')
5      st.markdown('<p class="big1-font">WordCloud</p>', unsafe_allow_html=True)
6      st.write(' ')
7      st.markdown(
8          '<p class="big1-font">Word clouds or tag clouds are graphical representations of
9          unsafe_allow_html=True)
10     st.write(' ')
11     st.write(' ')
12     long_string = ','.join(list(X_train1.values))
13     # Create a WordCloud object
14     wordcloud = WordCloud(background_color="white", max_words=5000, contour_width=3, con
15     # Generate a word cloud
16     wordcloud.generate(long_string)
17     # Visualize the word cloud
18     plt.figure(figsize=(20, 10))
19     plt.imshow(wordcloud)
20     st.image(wordcloud.to_array(), width=700)
21     st.write("Word Cloud")
22     wordcloud = WordCloud(width=3000, height=2000, random_state=1, background_color='bla
23                             collocations=False, stopwords=STOPWORDS).generate(long_string)
24     # Visualize the word cloud
25     plt.figure(figsize=(20, 10))
26     plt.imshow(wordcloud)
27     st.image(wordcloud.to_array(), width=700)

```

np7 hosted with ❤ by GitHub

[view raw](#)

Image by the author

5. Text Stat:

Textstat is an easy-to-use library to calculate statistics from the text. It helps determine readability, complexity, and grade level.

Some of the functions available in TextStat

- *Syllable Count*
- *Lexicon Count*
- *Sentence Count*
- *The Flesch Reading Ease formula*
- *The Flesch-Kincaid Grade Level*

- *The Fog Scale (Gunning FOG Formula)*
- *The SMOG Index*
- *Automated Readability Index*
- *The Coleman-Liau Index*
- *Linear Write Formula*
- *Dale-Chall Readability Score*
- *Readability Consensus-based upon all the above tests*

Check out Wikipedia for more info

The NY Times article can be read and understood by 6th and 7th-grade students.

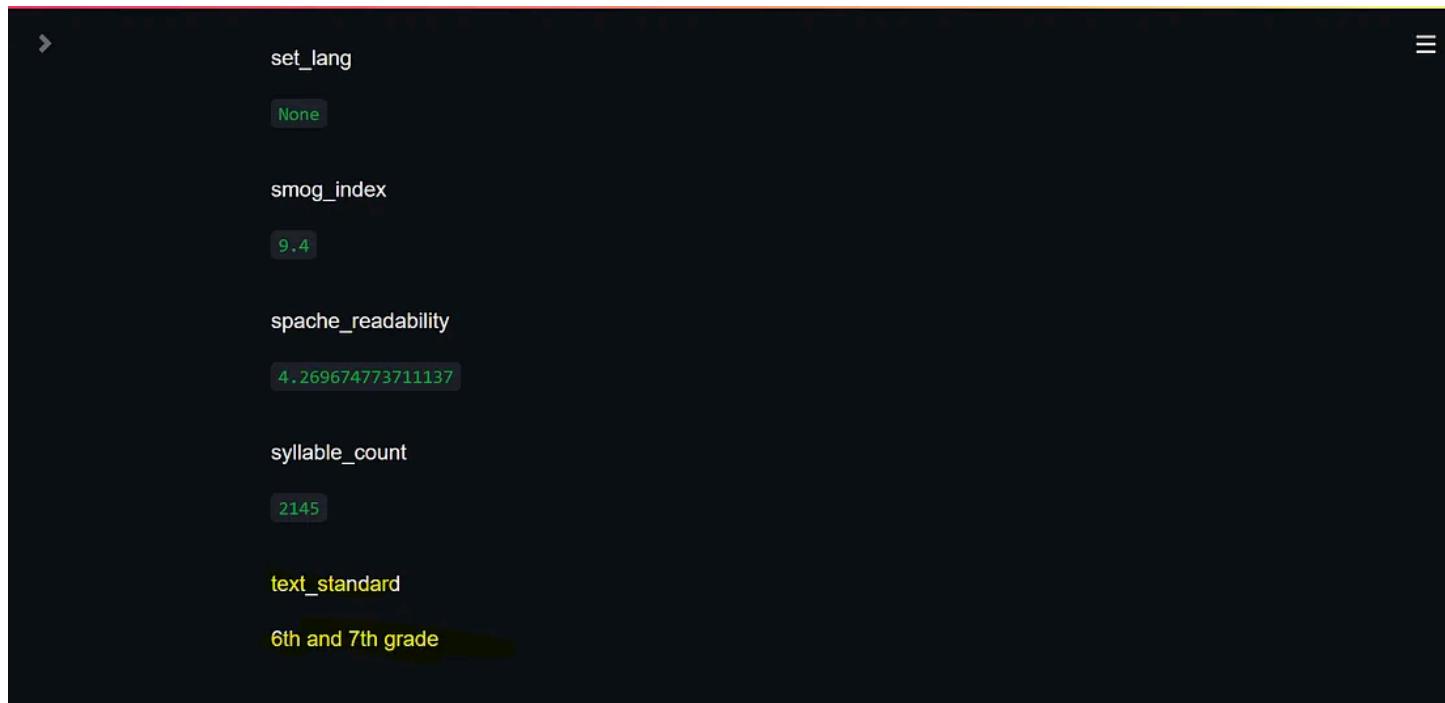


Image by the author — Screenshot of the NLP App

Some difficult words used in the article

```
1 : "assertive" ←  
2 : "behavior"  
3 : "trial"  
4 : "unresponsive"  
5 : "addiction"  
6 : "doing"  
7 : "approached"  
8 : "generally"  
9 : "caring"  
10 : "bogus"  
11 : "bystanders"  
12 : "identify"  
13 : "received"  
14 : "fentanyl"  
15 : "constricting"  
16 : "eventually"  
17 : "department"  
18 : "suggest"
```

Image by the author — Screenshot of the NLP App

The code to call the TextStat is below

```
1  if nlp == "Text Stat":  
2      import inspect  
3      import textstat  
4  
5  
6      st.markdown("""<style>.big-font {font-size:30px !important;}</style>""", unsafe_allow_html=True)  
7      textstat.set_lang("en")  
8      text =df['content'][2]  
9      funcs = ["textstat." + inspect.getmembers(textstat, predicate=inspect.ismethod)[i][0]  
10     st.write(" ")  
11     st.markdown('<p class="big-font">Textstat is an easy to use library to calculate sta  
12     st.write(" ")  
13     for elem in funcs:  
14         method = eval(elem)  
15         textstat.set_lang("en")  
16         w_1 = (elem.split(".")[1])  
17         st.write(w_1)  
18         st.write(method(text))  
19         st.write(" ")
```

np8 hosted with ❤ by GitHub

[view raw](#)

Image by the author — Screenshot of the NLP App

6. Topic Modeling:

Topic modeling is a type of modeling where you discover topics in a collection of documents.

For more information on topic modeling check this research paper.

I used LDA — Latent Dirichlet Allocation for topic modeling.

Topic Modeling used in a lot of applications and some are

- *Getting insights into customer support conversations*
- *Help in processing huge amount of data and tagging it with the correct topic*
- *Processing customer feedback*
- *Classifying customer tickets*
- *Detecting the urgency of the customers.*

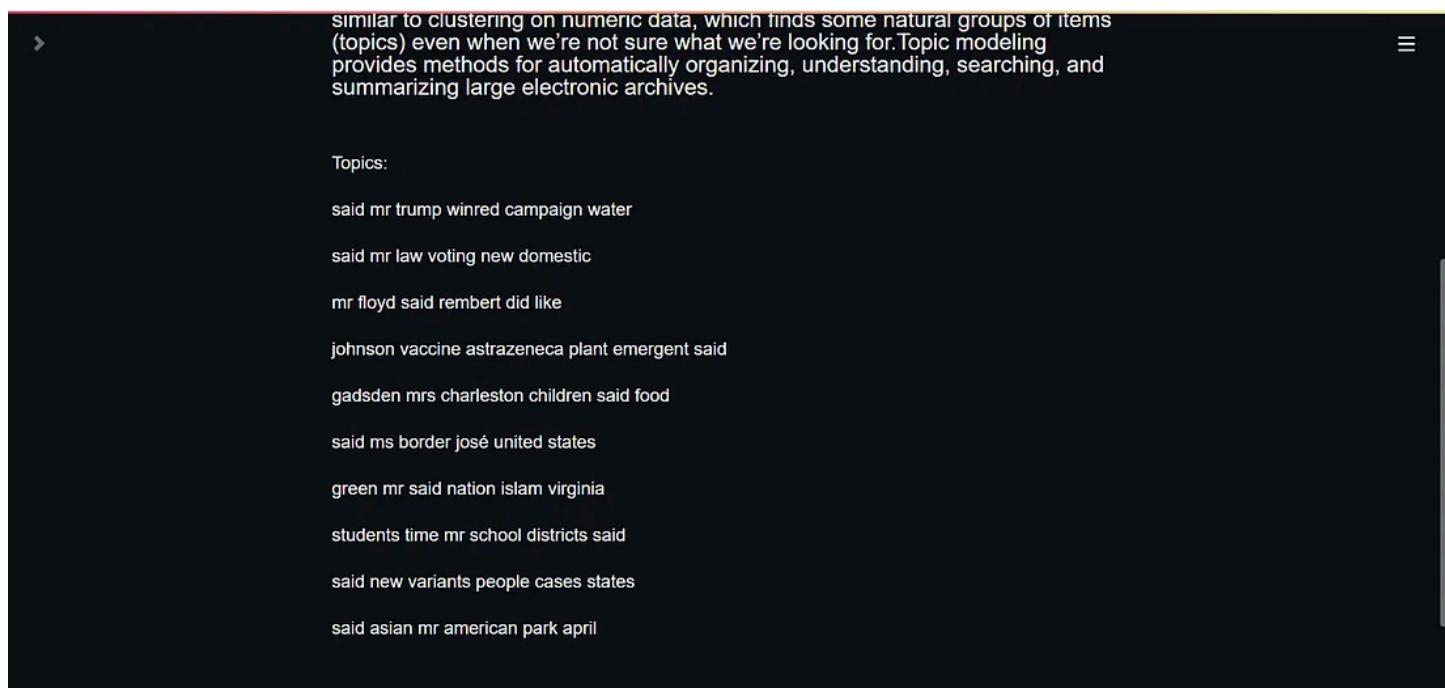


Image by the author — Screenshot of the NLP App

I think you can fine-tune it. Just try with the different parameters. The code is below

```

1  if nlp == "Topic Modeling":
2      st.write(" ")
3      st.write(" ")
4      st.header("Topic Modeling")
5      st.write(" ")
6      #st.write(" ")
7      st.subheader("Topic modeling is a method for unsupervised classification of document")
8      st.write(" ")
9      st.write(" ")
10     from sklearn.feature_extraction.text import CountVectorizer
11     import warnings
12     warnings.simplefilter("ignore", DeprecationWarning)
13     from sklearn.decomposition import LatentDirichletAllocation as LDA
14     def print_topics(model, count_vectorizer, n_top_words):
15         words = count_vectorizer.get_feature_names()
16         for topic_idx, topic in enumerate(model.components_):
17             #print("\nTopic #{}:".format(topic_idx))
18             w_stl = (" ".join([words[i]
19                               for i in topic.argsort()[-n_top_words - 1:-1]]))
20             st.write(w_stl)
21         number_topics = 10
22         number_words = 6
23         X_train1 = df['content']
24         count_vectorizer = CountVectorizer(stop_words='english')
25         count_data = count_vectorizer.fit_transform(X_train1)
26         lda = LDA(n_components=number_topics, n_jobs=-1)
27         lda.fit(count_data)
28         st.write("Topics:")

```

np9 hosted with ❤ by GitHub

[view raw](#)

Image by the author

7.Sentiment Analysis Using:

- *TextBlob*
- *Vader Sentiment Analyzer*

Sentiment analysis is the process of detecting positive or negative sentiment in text. It's often used by businesses to detect sentiment in social data, gauge brand reputation, and understand customers.

Sentiment Analysis is used in a lot of applications like the one below

- *Customer feedback*

- *Customer calls*
- *Customer tickets*
- *Social platform monitoring*
- *Help in market research*

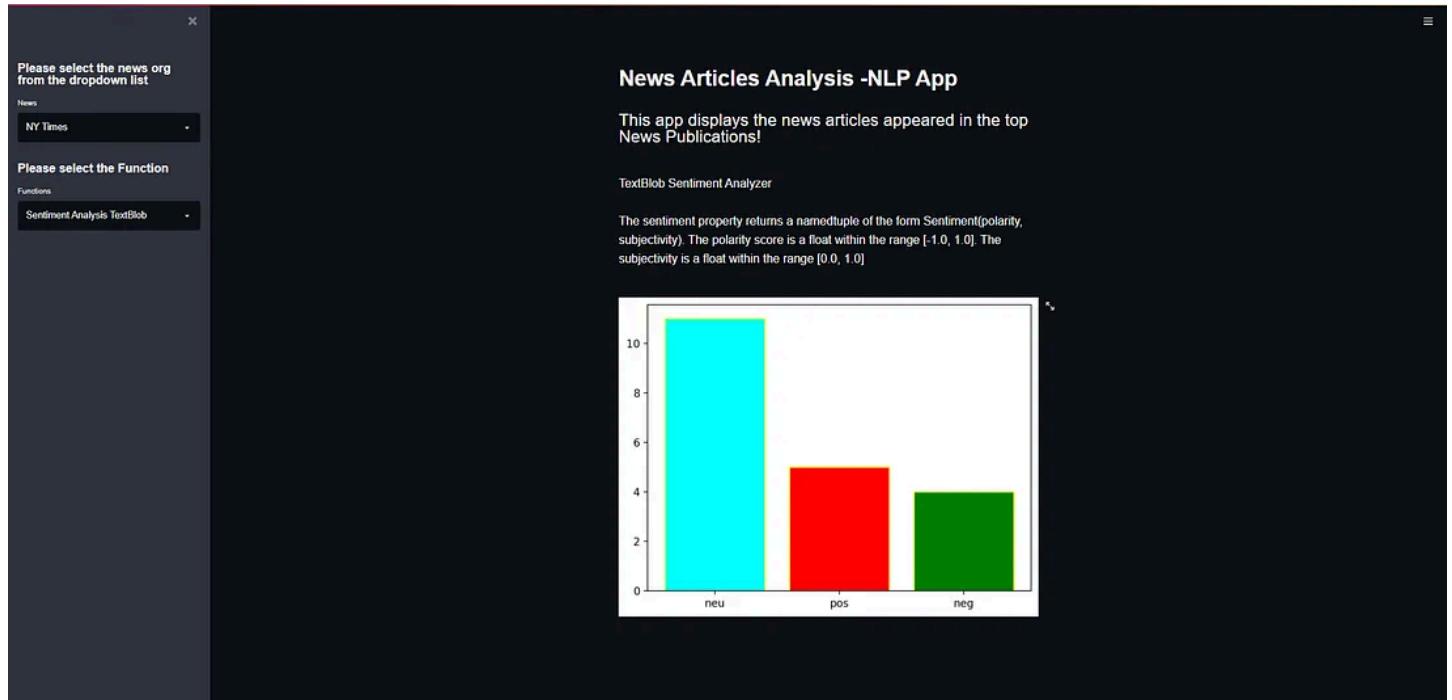
Textblob:

Image by the author — Screenshot of the NLP App

The code is below

```

1  @st.cache
2  def sentiment_vader(text, sid):
3      ss = sid.polarity_scores(text)
4      ss.pop('compound')
5      return max(ss, key=ss.get)
6  @st.cache
7  def sentiment_textblob(text):
8      x = TextBlob(text).sentiment.polarity
9      if x < 0:
10          return 'neg'
11      elif x == 0:
12          return 'neu'
13      else:
14          return 'pos'
15  #@st.cache(suppress_st_warning=True)
16  def plot_sentiment_barchart(text, method='TextBlob'):
17      if method == 'TextBlob':
18          sentiment = text.map(lambda x: sentiment_textblob(x))
19      elif method == 'Vader':
20          nltk.download('vader_lexicon')
21          sid = SentimentIntensityAnalyzer()
22          sentiment = text.map(lambda x: sentiment_vader(x, sid=sid))
23      else:
24          raise ValueError('Textblob or Vader')
25      plt.bar(sentiment.value_counts().index,
26              sentiment.value_counts(),color=['cyan', 'red', 'green', 'black'],edgecolor='yellow'
27      st.set_option('deprecation.showPyplotGlobalUse', False)
28      st.pyplot()
29
30  if nlp =="Sentiment Analysis TextBlob":
31      st.markdown("""
32                      <style>
33                      .big1-font {
34                          font-size:20px !important;
35                      }
36                      </style>
37                  """", unsafe_allow_html=True)
38  t_word = "The sentiment property returns a namedtuple of the form Sentiment(polarity
39  st.write(' ')
40  st.markdown('<p class="big1-font">TextBlob Sentiment Analyzer</p>',unsafe_allow_html
41  st.write(' ')
42  st.markdown('<p class="big1-font">The sentiment property returns a namedtuple of the
43  st.write(' ')
44  st.write(' ')
45  plot_sentiment_barchart(df['title'], method='TextBlob')
46
47  if nlp =="Sentiment Analysis-Vader":
48      st.markdown("""

```

```
49                                     <style>
50                                         .big1-font {
51                                             font-size:20px !important;
52                                         }
53                                     </style>
54                                     """", unsafe_allow_html=True)
55                                     st.write(' ')
56                                     st.markdown('<p class="big1-font">Vader Sentiment Analyzer</p>', unsafe_allow_html=True)
57                                     st.write(' ')
58                                     st.markdown(
59                                         '<p class="big1-font">VADER ( Valence Aware Dictionary for Sentiment Reasoning)
60                                         unsafe_allow_html=True)
61                                     st.write(' ')
62                                     st.write(' ')
63                                     plot_sentiment_barchart(df['title'], method='Vader')
```

Image by the author

8. Text Summarization:

Automatic summarization is the process of shortening a set of data computationally, to create a subset (a summary) that represents the most important or relevant information within the original content. Text summarization finds the most informative sentences in a document; image summarization finds the most representative images within an image collection; video summarization extracts the most important frames from the video content.

Text Summarization is used in a lot of applications like

- *Social Media Monitoring*
- *Legal Contract Analysis*
- *Financial/Equity Research*
- *Companies internal Document Analysis*
- *Teledoc – Medical Documents Summarization*
- *Books Summarization*
- *Patent Research*

[Check the Wikipedia definition.](#)

'I Was Failing': Bystanders Carry Guilt From Watching George Floyd Die

Had he not been busy, he said, he likely would have been the one to go out and confront Mr. Floyd and his friends about the fake bill, and the outcome might have been different.

He warned one of the officers that they too would be haunted by their actions, and that the officer would want to kill himself for what he did to Mr. Floyd.

As he spoke in court, it was clear that Mr. Williams was still struggling with what he saw.

Summarized Document

Fed Up With Remote Learning, Governors Make a Push to Reopen Schools

But when Mr. DeWine learned that Cleveland's school district was not planning to reopen until April, the governor said he told its chief executive, Eric Gordon, that he would not provide the vaccines if the district did not speed things up.

For much of this school year, Washington has had among the fewest students learning in person of any state, in part because it had imposed stringent requirements for schools to reopen.

Mr. DeWine's deal and Mr. Inslee's proclamation only required that districts offer students part-time in-person instruction.

Image by the author — Screenshot of the NLP App

The code is below

Open in app ↗



Search



```

1  if nlp == "Text Summarization":
2      st.write(" ")
3      st.write(" ")
4      st.header("Text Summarization")
5      st.write(" ")
6      # st.write(" ")
7      st.subheader(
8          "Text summarization refers to the technique of shortening long pieces of text. T"
9      st.write(" ")
10     st.write(" ")
11     from sumy.summarizers.lex_rank import LexRankSummarizer
12     #dfs = df['content']
13     for index, row in df.iterrows():
14         parser = PlaintextParser.from_string(row['content'], sumytoken("english"))
15         # Using LexRank
16         summarizer = LexRankSummarizer()
17         # Summarize the document with 4 sentences
18         summary = summarizer(parser.document, 3)
19         st.write("Summarized Document")
20         st.write(" ")
21         st.write(row['title'])
22         st.write(" ")
23         for sentence in summary:
24             #st.write("Summarized Document")
25             #st.write(row['title'])
26             st.write(sentence)

```

np11 hosted with ❤ by GitHub

[view raw](#)

Image by the author

9.Entity Extraction:

Named-entity recognition (NER) (also known as (named) entity identification, entity chunking, and entity extraction) is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, medical codes, time expressions, quantities, monetary values, percentages, etc.

[Check Wikipedia to read more about NER.](#)

NER is used in a lot of applications and some are

- *Product reviews*

- *In Search engine algorithms*
- *Email analysis*
- *Helping in indexing documents*
- *Customer service*



Image by the author — Screenshot of the NLP App

The code is

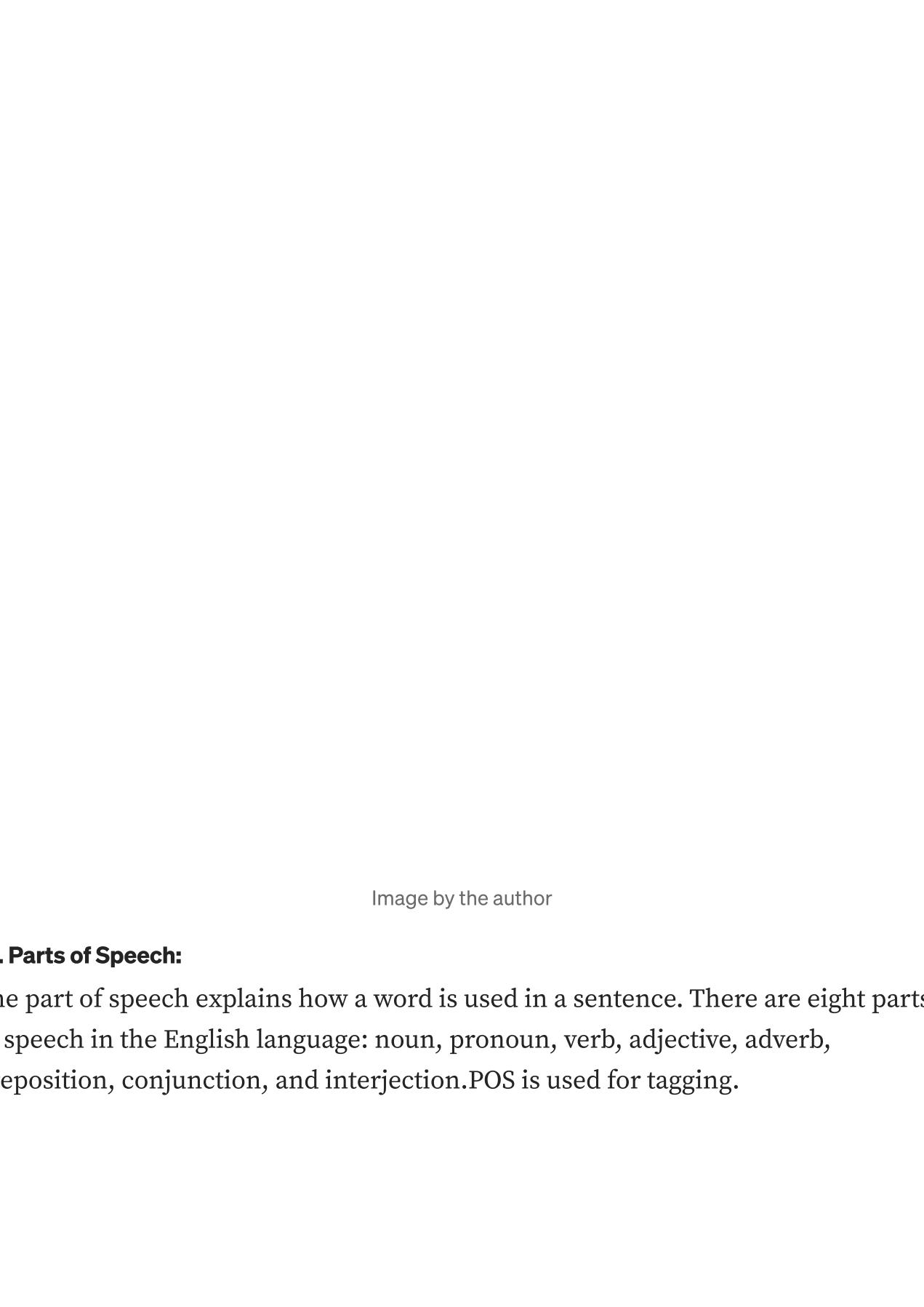


Image by the author

10. Parts of Speech:

The part of speech explains how a word is used in a sentence. There are eight parts of speech in the English language: noun, pronoun, verb, adjective, adverb, preposition, conjunction, and interjection. POS is used for tagging.

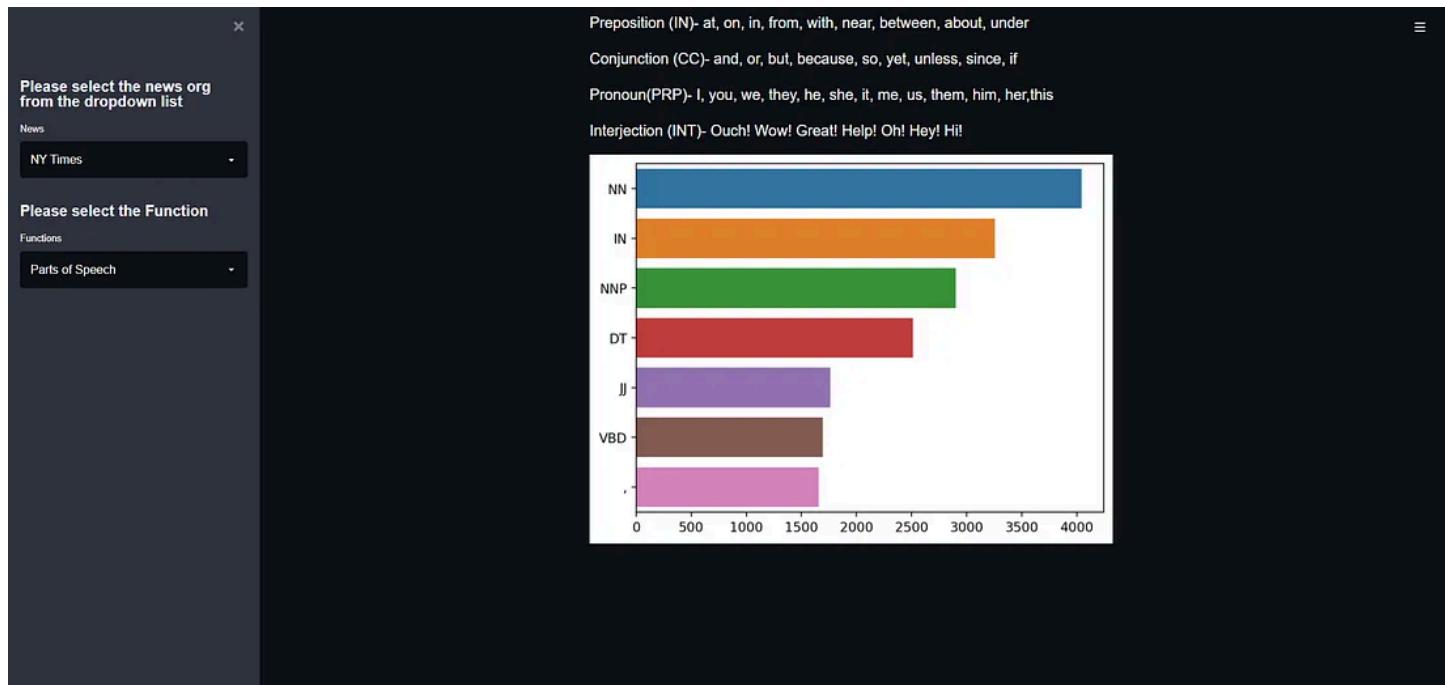


Image by the author — Screenshot of the NLP App

The code is



Image by the author

The gif file of the app:

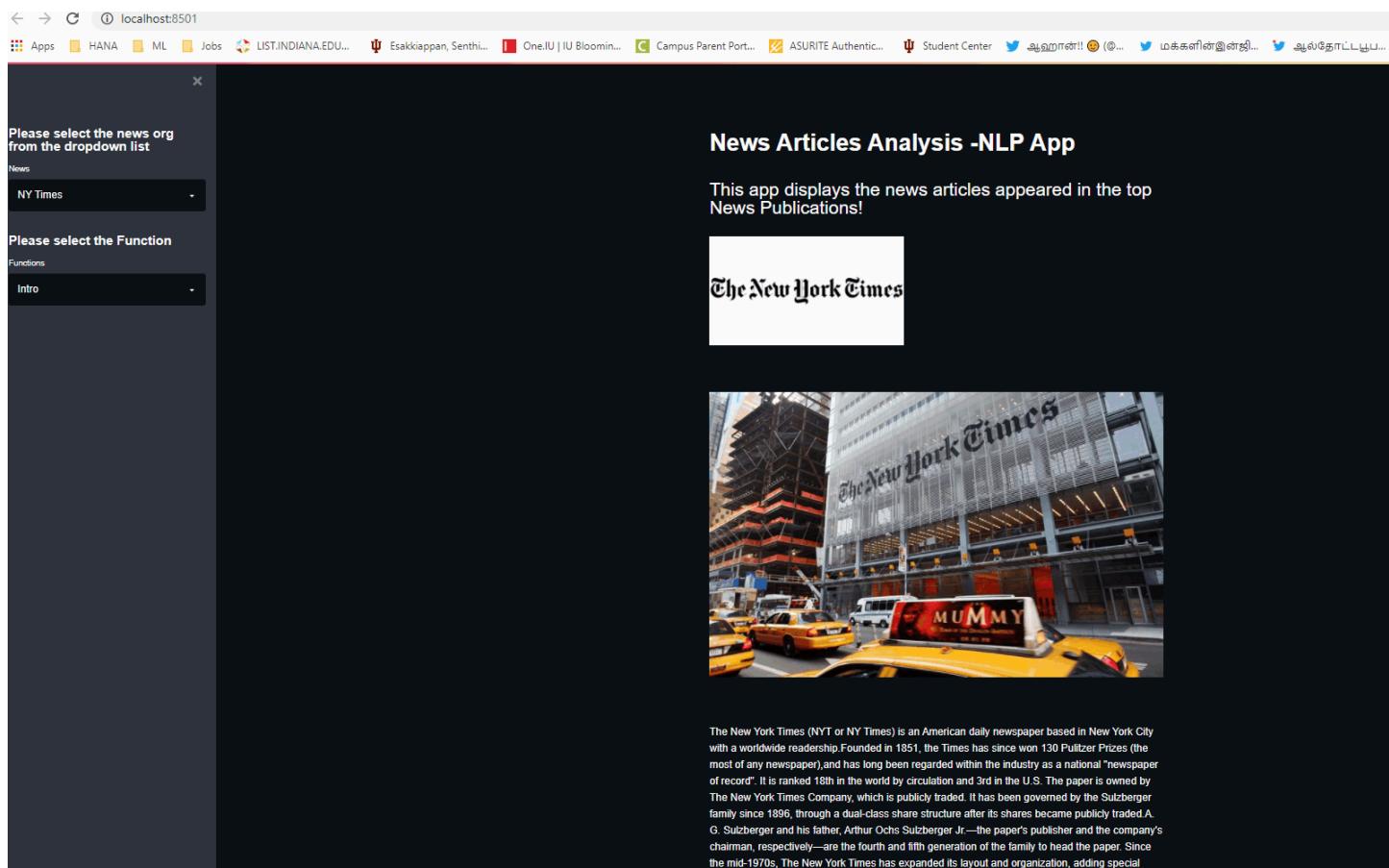


Image by the author — gif file of the NLP App

The last step is to deploy the app in Heroku. If you are interested in hosting your app then study further.

How to deploy the app in Heroku

Follow the steps to deploy the app in Heroku . There are a lot of articles on how to deploy an app in Heroku. Please check the below article which has the steps to deploy the app in Heroku

<https://towardsdatascience.com/from-streamlit-to-heroku-62a655b7319>

Also, check out the youtube video which walks through the deployment process

Credits to — Youtube video by DataProfessor

The files needed for deployment

- *Python script(The above python file with the logic)*
- *Procfile*
- *setup.sh*
- *requirements.txt*

The full code is

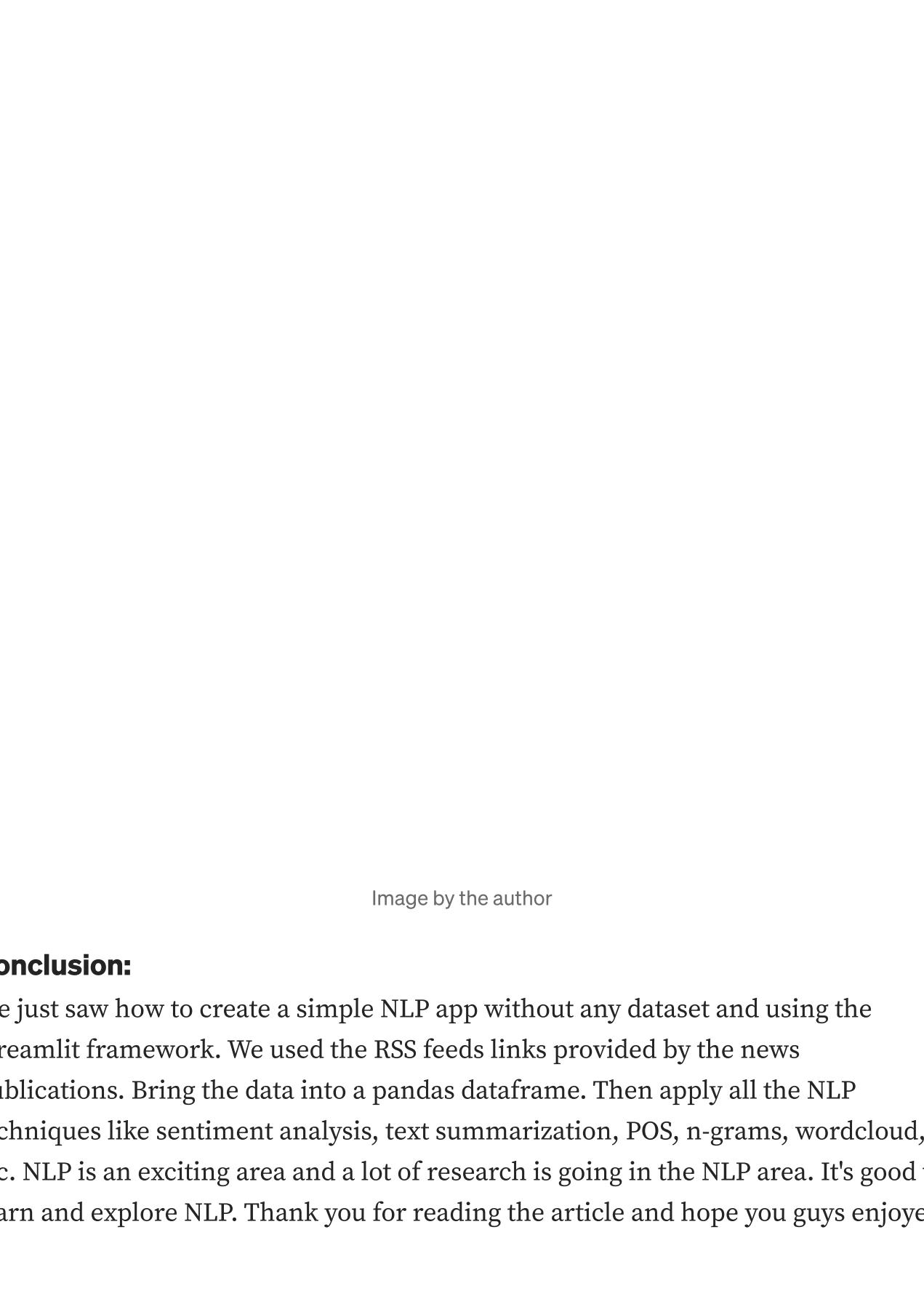
A small, faint watermark or logo in the background of the slide, possibly a camera lens or a similar circular emblem.

Image by the author

Conclusion:

We just saw how to create a simple NLP app without any dataset and using the Streamlit framework. We used the RSS feeds links provided by the news publications. Bring the data into a pandas dataframe. Then apply all the NLP techniques like sentiment analysis, text summarization, POS, n-grams, wordcloud, etc. NLP is an exciting area and a lot of research is going in the NLP area. It's good to learn and explore NLP. Thank you for reading the article and hope you guys enjoyed it.

Please feel free to connect with me on [LinkedIn](#)

References:

- [1]<https://streamlit.io/>
- [2]<https://monkeylearn.com/sentiment-analysis/>
- [3]<https://monkeylearn.com/blog/document-classification/>
- [4]https://en.wikipedia.org/wiki/Sentiment_analysis
- [5]<https://towardsdatascience.com/topic-modeling-and-latent-dirichlet-allocation-in-python-9bf156893c24>
- [6]<https://towardsdatascience.com/named-entity-recognition-in-nlp-be09139fa7b8>
- [7]<https://www.youtube.com/channel/UCV8e2g4IWQqK71bbzGDEI4Q>
- [8]<https://pypi.org/project/sumy/>
- [9]<https://pypi.org/project/textstat/>
- [10]<https://towardsdatascience.com/write-better-stories-with-this-python-tool-ad1371158822>
- [11]<https://textblob.readthedocs.io/en/dev/quickstart.html>
- [12]<https://pypi.org/project/vaderSentiment/>
- [13]<https://www.frase.io/blog/20-applications-of-automatic-summarization-in-the-enterprise/>
- [14]https://en.wikipedia.org/wiki/Named-entity_recognition
- [15]https://en.wikipedia.org/wiki/Part-of-speech_tagging

[NLP](#)[Machine Learning](#)[Deep Learning](#)[Data Science](#)[Python](#)[Following](#)

Written by Senthil E

2.7K Followers · Writer for Geek Culture

ML/DS - Certified GCP Professional Machine Learning Engineer, Certified AWS Professional Machine learning Speciality,Certified GCP Professional Data Engineer .

More from Senthil E and Geek Culture



 Senthil E in Level Up Coding

Navigating the World of LLMs: A Beginner’s Guide to Prompt Engineering-Part 2

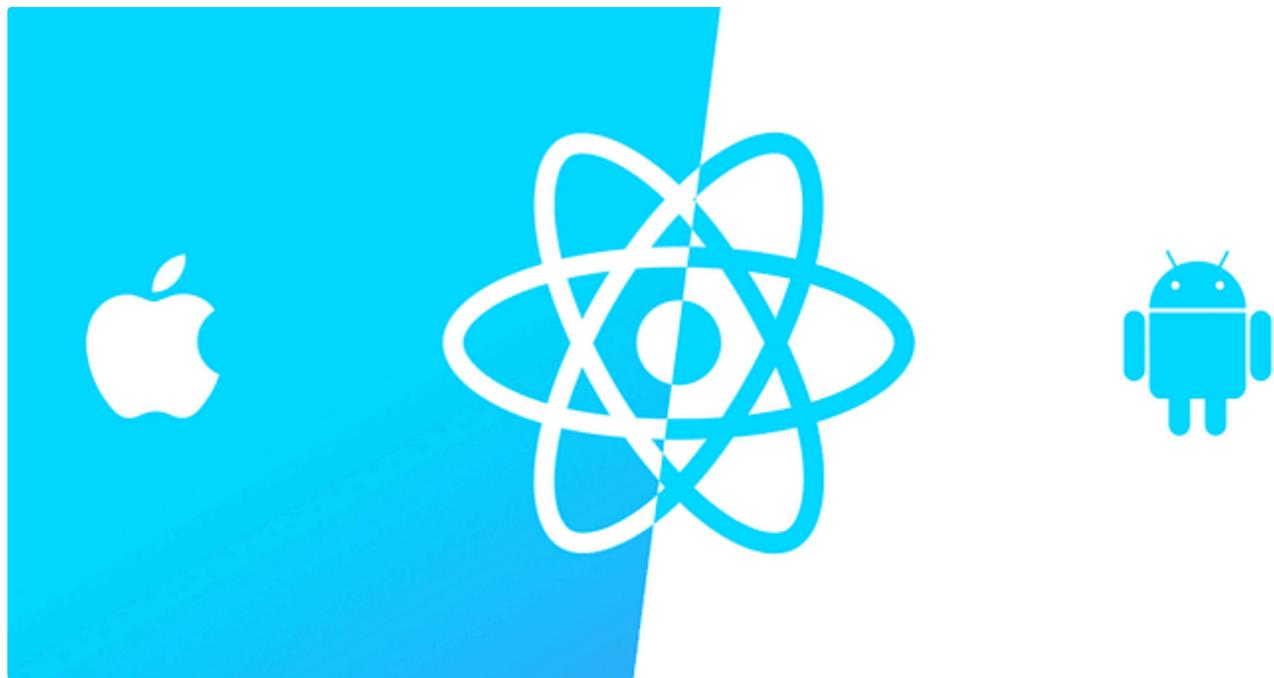
From Basics To Advanced Techniques

32 min read · Mar 17, 2024

 302



...



 Anshul Borawake in Geek Culture

React Native Generate APK—Debug and Release APK

Generate Debug and Release APK in React Native; Windows, iOS and Linux

3 min read · Apr 3, 2021

 1.7K  11



...

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System

Architecture



Version

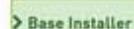
  

Installer Type

Download Installer for Windows 10 x86_64

The base installer is available for download below.



[Download \[2.8 GB\]](#) 

Installation Instructions:

 Techzizou in Geek Culture

Install CUDA and CUDNN on Windows & Linux

CONTENTS

8 min read · Aug 24, 2021

220

1



...



Senthil E in Level Up Coding

Unleashing the Potential of LLMs: How Enterprises are Leveraging AI for Enhanced Services

From Chatbots to Automation: Exploring the Versatile Use Cases of LLMs in Enterprises

58 min read · Mar 31, 2024

113



...

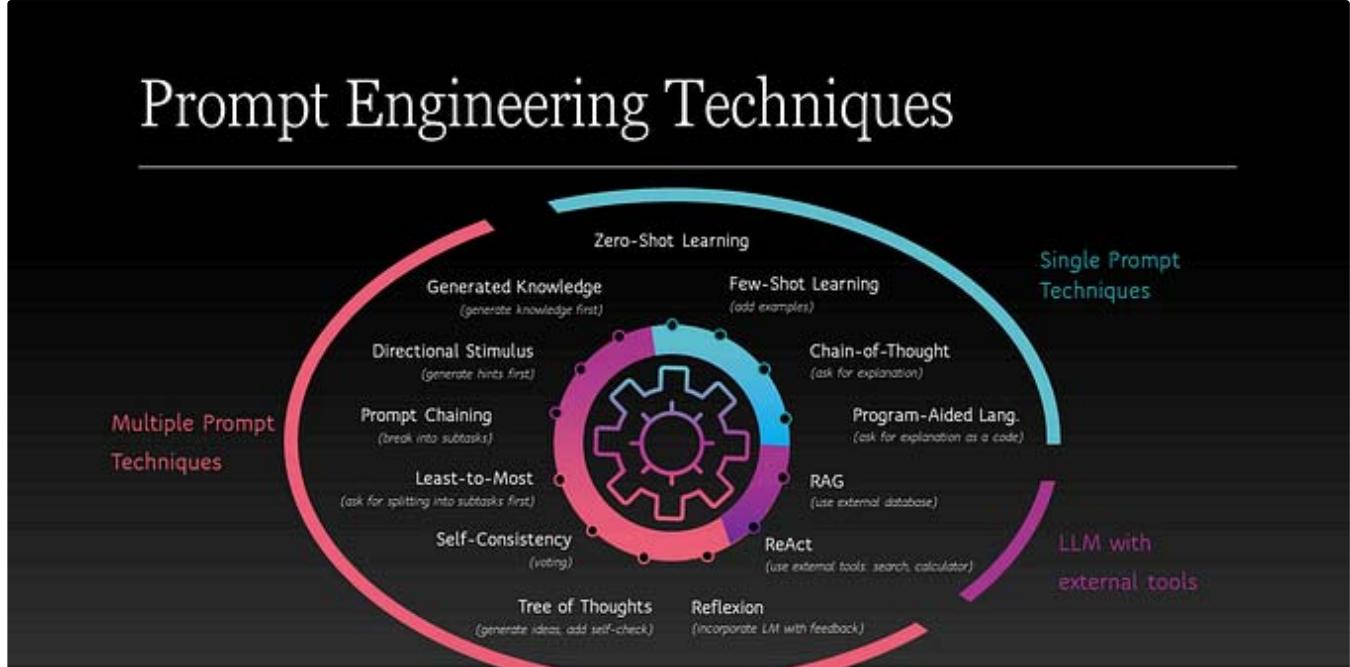
[See all from Senthil E](#)[See all from Geek Culture](#)

Recommended from Medium

<https://medium.com/geekculture/how-to-build-an-nlp-machine-learning-app-end-to-end-76404ea9f6e8>

36/40

Prompt Engineering Techniques



 Katsiaryna Ruksha

Prompt Engineering: Classification of Techniques and Prompt Tuning

Being an emerging field of study prompt engineering lacks definitive classification of techniques. When you look through different articles...

12 min read · Apr 12, 2024

 176  2



...



 Ameydhote

Fine-Tuning Language Models for NER: A Hands-On Step-by-Step Guide

With its unparalleled ability to comprehend and produce text that resembles that of a person, large language models, or LLMs, have become...

6 min read · Feb 1, 2024

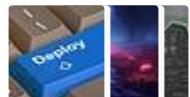
40

1

+

...

Lists



Predictive Modeling w/ Python

20 stories · 1111 saves



Practical Guides to Machine Learning

10 stories · 1323 saves



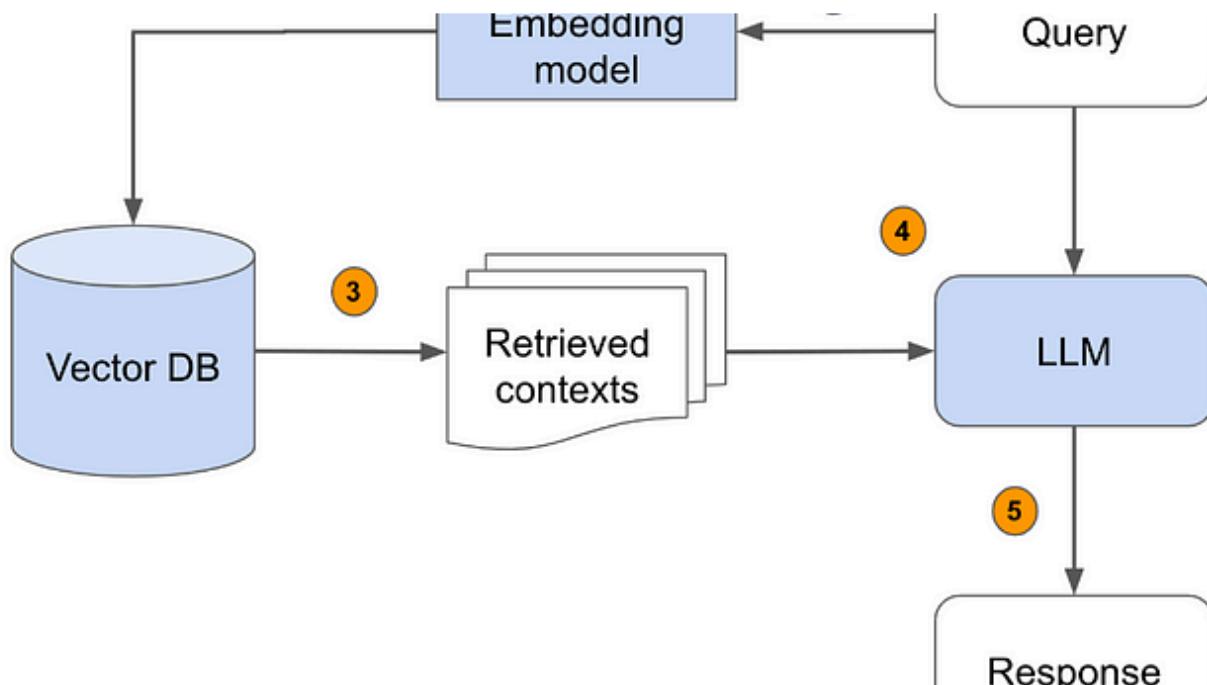
Natural Language Processing

1380 stories · 872 saves



Coding & Development

11 stories · 567 saves



Bijit Ghosh

RAG Vs VectorDB

Introduction to RAG and VectorDB

14 min read · Jan 28, 2024

187

4



...

HOW LARGE LANGUAGE MODELS WORK FROM ZERO TO CHATGPT

Andreas Stöffelbauer
Data Scientist @ Microsoft

Andreas Stöffelbauer in Data Science at Microsoft

How Large Language Models Work

From zero to ChatGPT

25 min read · Oct 24, 2023

1.2K

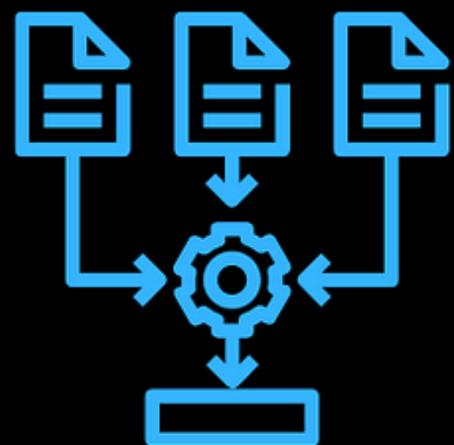
17



...

14 FREE NOTEBOOKS TO GET STARTED WITH

LARGE LANGUAGE MODELS FINE-TUNING





Youssef Hosni in Level Up Coding

14 Free Large Language Models Fine-Tuning Notebooks

Getting Started with LLM Fine-Tuning through These Free Colab Notebooks

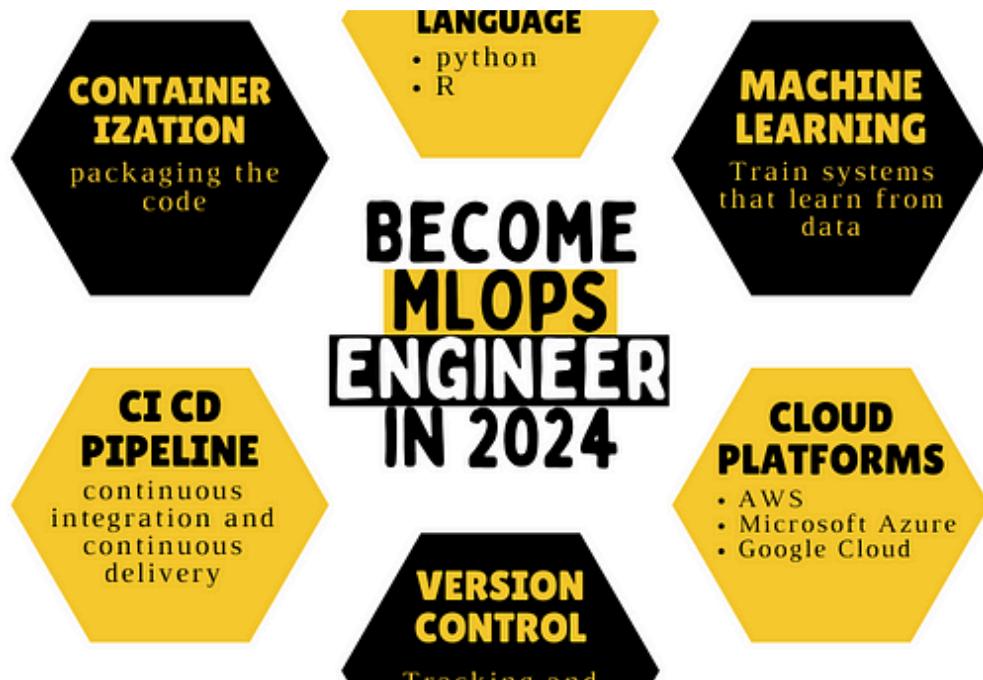
◆ · 10 min read · Feb 6, 2024

👏 649

💬 2



...



👤 Asad iqbal

MLOps Roadmap | How To Become MLOps Engineer in 2024

A Comprehensive MLOps roadmap to become MLOps engineer in 2024

8 min read · Apr 1, 2024

👏 87

💬 1



...

See more recommendations