Hochschule für Technik und Wirtschaft Berlin
University of Applied Sciences

Extraction, Keywording and Clustering of Texts Highly Relevant to Social Media Discourses on COVID-19

Project Report

Project Studies on Contemporary Topics

Professional IT Business and Digitalization

Date: 12/05/2022

Submitter: Shabbir Ahmad Farooquee

Supervisor: Prof. Dr. Helena Mihaljević

Abstract

COVID-19 spread rapidly across the globe, at the same time it has been observed that the misinformation and conspiracy theories related to COVID-19 spread even faster. Some studies show that the belief in conspiracy and misinformation are related to anxiety and depression and even protests and violence. Most of the misinformation and conspiracy are debunked and proven false but it kept shared and believed on social media platforms. The reason is social media platforms and digital technologies facilitate high-speed misinformation sharing between news producers and consumers as well as cross-platform information cascades [1]. With the integration of machine learning and text-based processing, we built classifiers that can classify the news data. Text classification mainly focuses on extracting various features of text and after that incorporating those features into classification. We used Newspaper3k and Gensim libraries for text processing and extracting keywords and compared the results. These two libraries work effectively in general, but Newspaper3k worked better for this project because all texts are in German. The extracted keywords are converted to vectors using the word2vec model and the vector average is calculated for each news article. Considering vector average, news articles are classified using the unsupervised machine learning algorithm “K-means clustering”. As a result, all news articles are clustered into two groups where each group contains similar news articles.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

The COVID-19 situation has created many challenges and one of the main challenges faced by us is the spread of misinformation and conspiracy theories related to it. The misinformation spreads very fast through social media where news articles are shared from different sources. The big challenge in this area is the lack of an efficient way to differentiate between fake and non-fake news due to the unavailability of corpora. In this project, news articles are taken from different social media platforms to extract the content and classify based on keywords. Classification will help to identify the social media posts which are spreading misinformation. To achieve this goal, the most popular news websites are chosen along with the URLs. The URLs are accessed to extract the title, text, summary, and keywords for each news article in the dataset using natural language processing methods. Two Python libraries Newspaper3k and Gensim are considered, and results are compared. Newspaper3k supports many languages including German and gives better results, so it is preferred over Gensim for this project where all texts are in German. After adding keywords to the dataset for each news article, data-cleaning methods are applied. The keywords in the dataset are considered for the classification of news articles, which are converted to vectors to be consumed by the unsupervised machine learning algorithm K-means clustering. To structure this project, the CRISP-DM model is followed which is defined for data mining projects. It is adapted for unsupervised machine learning, as data mining and machine learning share many approaches and procedures. At each stage, steps are selected based on this specific project and the data collected.

The CRISP-DM (Cross Industry Standard Process for Data Mining) defines a comprehensive process model to execute data mining projects. As a methodology, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks shown in Figure 1 [2].

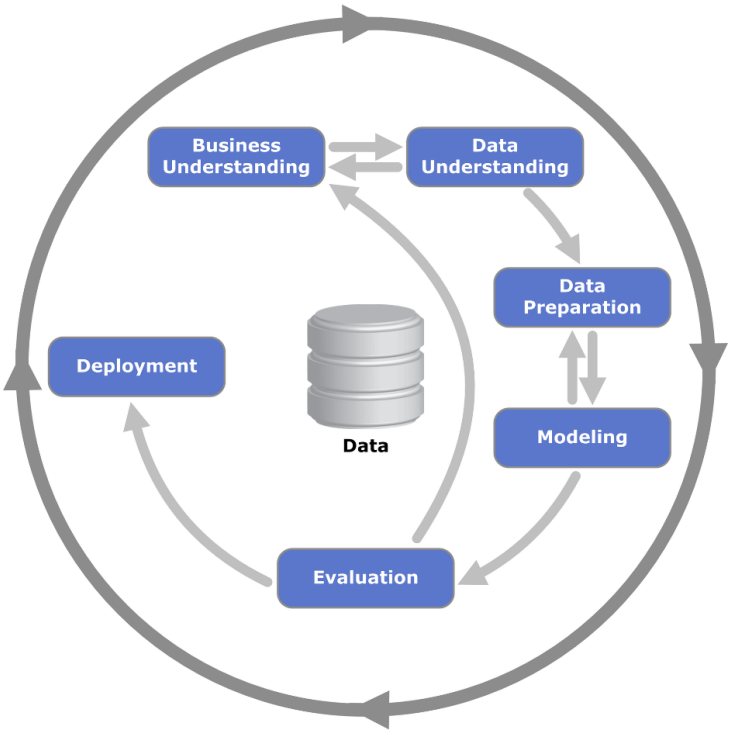

Figure 1. 1: CRISP-DM model

This model consists of six iterative phases from business understanding to deployment as shown in the above figure (Figure-1). In the Business Understanding phase, the business situation is supposed to be assessed to get an overview of the available and required resources. The determination of the data mining goal is one of the most important aspects of this phase. First, the business background information should be explained. The definition of business objectives and business success criteria are other important aspects of this phase. In the Data Understanding phase, the essential tasks are collecting data from data sources and exploring and describing it. Data quality is also determined in this phase along with data integration. In the Data Preparation phase, data quality is improved by inclusion or exclusion criteria. Quality is further improved by the process of data cleaning by removing bad data. In the Modeling phase, the modeling technique is selected based on the business problem and the dataset. Evaluation criteria are considered while selecting the model and all model-based parameters are defined at this stage. In the evaluation phase, the results are checked against the defined business objectives. Therefore, the results must be interpreted and further actions have to be defined. Also, the process should be reviewed in general [3].

Chapter 2: Business Understanding

The spread of COVID-19 has changed every aspect of life to such an extent that everybody is affected by it. The situation has disrupted day-to-day life and everybody is concerned and talking about it on all platforms. On social media, the information is shared and spread quickly. The information shared is from reliable news sources which are written based on facts and figures. On the other hand, social media users also shared links from dodgy websites which is spreading rumors and creating unrest based on narrative.

To differentiate the reliable and dodgy websites, the news articles shared on social media can be collected and clustered using natural language processing and unsupervised machine learning. A trained model should be able to extract the main content from a news article shared on social media, extract keywords based on text extracted, and cluster related articles based on the keywords.

2.1: Business Objectives

In this study project, the content of frequently linked news websites will be extracted and tagged by keywords. Specifically, this requires the following work steps:

1. Extraction of URLs from social media posts and finding the most frequent ones that are not typical video hosting platforms. The main focus is on news websites.
2. Application of Python libraries Newspaper3k and Gensim to extract the main content from news articles.
3. Automatic extraction of summary and keywords from the cleaned texts using Natural Language Processing and Machine Learning Algorithms.
4. Converting keywords to word vectors and clustering all the articles using an unsupervised machine learning algorithm.

2.2: Business Success Criteria

The extraction of the main text in each article is one of the main tasks to be performed by applying natural language processing and machine learning algorithms. Once the main text is

extracted, it will be considered input for extracting summary and keywords. Machine learning algorithms can understand only numbers, so all keywords should be converted to vectors using predefined Python libraries. Then, the average of vectors is calculated for clustering of all news articles using an unsupervised machine learning algorithm. The following tasks are performed to complete this project:

- Extraction of text and keywords for each article (after data cleaning)
- Conversion of each keyword into a vector
- Calculation of vector average, which should be done by adding all vectors for a particular news article and dividing by the total number of vectors for that news article
- Clustering of news articles based on vector average

Chapter 3: Data Understanding

Social media users are sharing COVID-19 related news from a large number of domains. The URLs shared are tweets, videos, news links, etc. Each link is added to the dataset as a record along with the domain name and tag in separate columns. These shared links can be categorized broadly as follows:

- post from social networks (Twitter, Facebook, etc)
- Video links from video-sharing platforms (youtube.com)
- News articles from various news websites (welt.de, tagesspiegel.de etc)
- A long list of dodgy websites

The dataset has 3 columns and 401,411 records. The columns are as follows:

- expanded_url
- domain
- tag

Further analysis of the dataset shows that there are a total of 9,948 unique domains. Some domains are popular (e.g.: twitter.com, youtube.com, etc) and others are not shared as frequently as Twitter and YouTube (e.g.: news websites). The top 10 domains in the dataset are shown in Table 3.1. It has been observed that the dodgy websites are not posted frequently and roughly two-thirds of the records are not shared more than once.

Table 3. 1: List of top 10 domains

Domain	No of records
twitter.com	288363
youtu.be	14964
youtube.com	10339
welt.de	1836
tagesspiegel.de	1804
tagesschau.de	1799
de.wikipedia.org	1555
spiegel.de	1483
zeit.de	1179
facebook.com	1169

In this project, the Python pandas library is used to understand the dataset. Pandas function "describe()" is used for a descriptive statistical summary of all the features. Also, the "dtypes()" function is used to check the type of each variable in the dataset. The following output is displayed in the Jupyter Notebook:

Import libraries

In [1]:
```
import pandas as pd
import numpy as np
import os
import pandas_profiling
```

Import dataset

In [2]:
```
file_path = os.path.join("data", "twitter_URLs.bin")
df = pd.read_csv(file_path)
df.head()
```

Out[2]:

	expanded_url	domain	tag
0	https://twitter.com/kjh_mov/status/12996294579...	twitter.com	NaN
1	https://twitter.com/kjh_mov/status/12996294579...	twitter.com	NaN
2	https://twitter.com/kjh_mov/status/12996294579...	twitter.com	NaN
3	http://ntv.de	ntv.de	NaN
4	https://twitter.com/kjh_mov/status/12996294506...	twitter.com	NaN

Figure 3. 1: Import libraries and dataframe

In [3]:
```
df.describe()
```

Out[3]:

	expanded_url	domain	tag
count	401411	401411	27904
unique	286502	9948	9
top	https://support.twitter.com/articles/20169199	twitter.com	Youtube
freq	623	288363	25311

In [4]:
```
df.dtypes
```

Out[4]:
```
expanded_url    object
domain          object
tag             object
dtype: object
```

Figure 3. 2: Output displayed by df.describe() and df.dtypes

```
In [5]: df["domain"].value_counts()

Out[5]: twitter.com                 288363
        youtu.be                     14964
        youtube.com                  10339
        welt.de                       1836
        tagesspiegel.de               1804
                                      ...
        dion-arts.com                    1
        grundschule-gartnisch.de         1
        griechenlandsoli.com             1
        dt.parteienlandschaft.so         1
        beta.ctvnews.ca                  1
        Name: domain, Length: 9948, dtype: int64
```

Figure 3. 3: List of domains

Pandas “profile-report()” is used to generate a report [4]. Summary of profile-report are as follows:

- dataset has 1,14,909 (28.6%) duplicate values
- expanded_url has 2,86,502 (71.4%) distinct values
- domain column has 9948 (2.5%) distinct values
- tag column has 3,73,507 (93.0%) missing values

Chapter 4: Data Preparation

4.1: Selecting Top News Websites

There are high numbers of domains in the dataset. In this project, the focus is on news websites. After reviewing the most frequent news websites, I have selected the top 5 for data preparation. The top 5 selected news websites are the following (Table 4.1):

Table 4. 1: List of top 5 news websites

Selected News Website	No of records
welt.de	1836
tagesspiegel.de	1804
tagesschau.de	1799
spiegel.de	1483
zeit.de	1179

Figure 4. 1: Code snippet to select top 5 news websites

4.2: Drop duplicate records

There are a total of 8101 records from the top 5 news websites (as shown in Figure 4.1). It also shows that 3804 URLs are unique and the remaining are duplicates. The duplicate URLs are dropped as shown in Figure 4.2. As a result, 4297 duplicate records are dropped and the dataframe is reduced to 3804 records.

2.2 drop duplicate rows and reset index

In [8]:
```
df = df.drop_duplicates()
df = df.reset_index(drop=True)
df.describe()
```

Out[8]:

	expanded_url	domain	tag
count	3804	3804	0
unique	3804	5	0
top	https://www.spiegel.de/wissenschaft/der-einflu...	spiegel.de	NaN
freq	1	848	NaN

Figure 4. 2: Drop duplicate records and reset index

4.3: Extract Title, Text, Summary and Keywords

It is required to access each URL in the dataframe and scrape through the news articles to extract the main content. There are Python libraries available for scraping news articles, but language support (with option "de") must be considered here since all texts are in German. Newspaper3k has been used to extract the title and main text. After that, the main text from the news articles can be used by Newspaper3k and Gensim to extract summaries and keywords. These 2 results from Newspaper3k and Gensim are compared in the next step.

4.3.1: Newspaper3k

Newspaper3k library is created and maintained by Lucas Ou-Yang. It is powered by "lxml" [5] and its simplicity is inspired by another Python library called "request" [6]. This library uses a lot of "python-goose" parsing code [7]. Newspaper3k has seamless language extraction and detection. It supports a long list of languages, if no language is specified Newspaper will attempt to auto-detect it. The following list of features are supported by this library [8]:

- Multi-threaded article download framework
- News URL identification
- Text extraction from HTML
- Top image extraction from HTML
- All image extraction from HTML
- Keyword extraction from text
- Summary extraction from text
- Author extraction from text
- Google trending terms extraction

- Works in 10+ languages (English, German, Chinese, Arabic etc)

To utilize the above features, an article should be initialized and the following methods should be called with the initialized object:

- article.download()
- article.parse()
- article.nlp()

The first method downloads the news article. It keeps the HTML of the article's body text and retains the semantic information of HTML. The DOM object is cached for the user for further operations [8]. After downloading, the parse() method can be called to extract the title, author, text, etc. Finally, the nlp() method should be called to extract natural language properties (summary and keywords). All these 3 methods must be called in the order because the parse() method has a dependency on download(), then nlp() has a dependency on download() and parse() [9].

To use this library, a correct version is installed based on the Python version ("pip3 install newspaper3k"). To extract details about the article, a class named "Article" is defined in the library. An object should be created for the "Article" class by passing 2 parameters (URL and language). The URL is taken from the dataframe column "expanded_url" and the language parameter is set as language= "de". The language parameter is important because this library supports a long list of languages. The "Article" method has been executed in the loop to create objects for each news article in the dataset. Other methods defined in the library are called for downloading, parsing, and natural language processing. Then, relevant parameters (title, text, summary, and keyword) are called and saved in the dataframe. To achieve this task, new columns are created in the dataframe (title, text, summary, and keywords) and all steps are executed in a loop to cover all URLs in the dataset. Some URLs are no longer active, in this case, downloading/parsing will fail with an exception. These exceptions are handled and error messages are printed along with the article number.

2.3 extract title, text, summary and keywords

pip3 install newspaper3k

```
In [9]: from newspaper import Article

        #create new columns for article title, text, summary and keywords
        df["title"] = np.nan
        df["text"] = np.nan
        df["summary"] = np.nan
        df["keywords"] = np.nan

        #extract title, text, summary and keywords for each article and copy to corresponding row-column
        url_list = df.expanded_url
        for i in range(0,len(url_list)):
            print("Article number: "+str(i))
            try:
                article = Article(url_list[i], language="de")
                article.download()
                article.parse()
                article.nlp()

                df["title"][i] = article.title
                df["text"][i] = article.text
                df["summary"][i] = article.summary
                df["keywords"][i] = article.keywords
            except Exception as e:
                print(e)
```

Figure 4. 3: Code snippet for extracting title, text, summary, and keywords

4.3.2: Gensim

Gensim is a Python library that is open source and can be used to represent documents as semantic vectors. It uses unsupervised machine learning algorithms and is designed to process raw and unstructured plain digital texts. The algorithms used in Gensim automatically discover the semantic structure of documents by examining statistical co-occurrence patterns within a corpus of training documents. These algorithms are unsupervised, which means no human input is necessary, only a corpus of plain text documents is required. Once these statistical patterns are found, any plain text documents (sentence, phrase, word, etc) can be succinctly expressed in the new, semantic representation and queried for topical similarity against other documents (words, phrases, etc) [10].

After extracting text from news articles, the Gensim library is used to extract summaries and keywords. There are functions defined in this library called “summarize()” and “keywords()”, which take multiple parameters along with the extracted text.

gensim.summarization.summarize(text, ratio=0.15, split=True)

For summarize(), it takes the article as one parameter and the ratio parameter determines the number of sentences in the summary text which is used to determine its size/length. The third parameter split=True means it will return a list of sentences.

gensim.summarization.keywords(*text, words=10, split=",", lemmatize=False*)

For keyword function(), the word parameter is used to determine the number of keywords. In this case, the function returns a list of keywords separated by "," for each article. By default, words are returned in inflected forms which can be handled by setting lemmatize=True. When lemmatize=True is set, the function returns the exact number of words defined in the function parameter and the list of keywords is lemmatized. On the other hand, when lemmatize=False, it is observed that the function returns more keywords than expected. The reason is that all inflected words are grouped together and counted as one.

Gensim

```
import gensim

def gensim_summarization(corpus, ratio=0.2):
    if type(corpus) is str:
       corpus = [corpus]
    summary = [gensim.summarization.summarize(text, ratio=ratio) for text in corpus]
    return summary
```

```
df["gen_summ_10"] = np.nan
df["gen_summ_15"] = np.nan
df["gen_summ_20"] = np.nan

for i in range(20):
    text_summ_i = gensim_summarization(df["text"][i], ratio=0.1)
    df["gen_summ_10"][i] = text_summ_i

for i in range(20):
    text_summ_i = gensim_summarization(df["text"][i], ratio=0.15)
    df["gen_summ_15"][i] = text_summ_i

for i in range(20):
    text_summ_i = gensim_summarization(df["text"][i])
    df["gen_summ_20"][i] = text_summ_i
```

Figure 4. 4: Code snippet for summarization using Gensim library

```
from gensim.summarization import keywords

df["gen_keywords"] = ""

for i in range(20):
    keywords_i = keywords(df["text"][i], ratio=0.2, split=",")
    df["gen_keywords"][i] = keywords_i
```

Figure 4. 5: Code snippet for keyword extraction using the Gensim library

It has been observed that there is no option to set language when using the Gensim library for text extraction or processing. This affects the quality of the summary and keywords when all texts are in German. On the other hand, Newspaper3k supports many languages (including German) and we can set language= “de” when calling the function. The quality of the summary and keywords extracted is better with Newspaper3k. Due to the above reason, Newspaper3k is preferred for this project.

4.4: Remove records with empty keywords field

The title, text, summary, and keywords are extracted and saved in the dataframe in the previous steps. It is required to remove null values from the dataframe as part of the data cleaning process. I have removed all the records where the keyword field is null (as shown in Figure 4.6). As a result, the dataset is reduced from 3804 records to 3776 records which means there were 28 records with null values in the keywords column.

2.4 removing record if keywords field is null

In [12]:
```
#remove all rows which has no keywords and reset index
df = df[~df['keywords'].isnull()]
df = df.reset_index(drop=True)
df.describe()
```

Out[12]:

	expanded_url	domain	tag	title	text	summary	keywords
count	3775	3775	0	3775	3775	3775	3775
unique	3775	5	0	2126	2101	2085	2197
top	https://www.spiegel.de/wissenschaft/der-einflu...	spiegel.de	NaN	Lesen Sie zeit.de mit Werbung oder im PUR-Abo....	zeit.de mit Werbung\n\nBesuchen Sie zeit.de wi...	zeit.de mit WerbungBesuchen Sie zeit.de wie ge...	[zeitde, wahl, lesen, werbung, werbungbesuchen...
freq	1	838	NaN	640	640	640	640

Figure 4. 6: Code snippet for removing null values in keywords field

4.5: Remove Records having PDF/Image as URLs

The dataframe has some web links for pdf and image files. The text cannot be extracted from PDF and image files using the Newspaper3k library, resulting in empty records in such cases. This situation is handled by removing such records. This operation dropped 16 rows and the dataframe is reduced to 3760 records as shown in Figure 4.7.

2.5 removing rows with .pdf/.jpg/.png links

In [15]:
```
#add code for removing pdf links
df_bkp_2 = df.copy()

df=df[~((df.expanded_url.str.endswith(".pdf")) | (df.expanded_url.str.endswith(".jpg")) | (df.expanded_url.str.endswith(".png"))]
#reset index
df = df.reset_index(drop=True)
df.describe()
```

Out[15]:

	expanded_url	domain	tag	title	text	summary	keywords
count	3760	3760	0	3760	3760	3760	3760
unique	3760	5	0	2125	2094	2079	2191
top	https://www.welt.de/debatte/kommentare/plus215...	spiegel.de	NaN	Lesen Sie zeit.de mit Werbung oder im PUR-Abo....	zeit.de mit Werbung\n\nBesuchen Sie zeit.de wi...	zeit.de mit WerbungBesuchen Sie zeit.de wie ge...	[zeitde, werbung, purabo, tracking, lesen, gew...
freq	1	838	NaN	639	639	639	639

Figure 4. 7: Removing records with PDF and image files

4.6: Load word2vec Library

The dataset is ready with a set of keywords for each news article, and news articles can be classified based on these keywords. However, machine learning algorithms cannot understand text, so all keywords should be converted to numbers (vectors in this case). In this project, the word2vec algorithm is considered which is created by a research team led by Tomas Mikolov at Google. It uses a neural network model to learn word associations from a large corpus of texts. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space [11].

For this project, the Gensim library is used because it is the fastest library for the training of vector embedding. The core algorithm of Gensim is highly optimized and Parallelized C routines [12]. To proceed, pre-trained word vectors are required which can be downloaded from the following link: https://dl.fbaipublicfiles.com/fasttext/vectors-wiki/wiki.de.vec

The downloaded model is trained on Wikipedia using fastText (a library created by Facebook AI Research Lab for learning word embeddings and text classification). These vectors in dimension 300 were obtained using the skip-gram model described in Bojanowski et al. (2016) with default parameters [13] [14].

The module “models.keyedvectors” from the Gensim library is used to load the model (Figure 4.8). Before this, the model is downloaded and saved locally. The loading process may take a longer time depending on system hardware configuration.

2.6 load library - word2vec

In [ ]:
```
# !pip install --upgrade gensim
```

In [17]:
```
#load word2vec
from gensim.models import KeyedVectors
vecs = KeyedVectors.load_word2vec_format('data/wiki.de.vec', binary=False)
```

Figure 4. 8: Load word2vec model using Gensim library

4.7: Calculate Vector Average

Every news article in the dataset has a list of keywords. These keywords are converted to vectors using the word2vec model. For every article, the sum of all vectors is calculated and then divided by the number of vectors. This gives the vector average for each article which is stored in a new column in the dataset. While calculating the average, a keyword is ignored if it is not found in the vector model.

2.7 calculate vector average

In [18]:
```
df["vec_sum_avg"] = np.array

for i in range(len(df["vec_sum_avg"])):
    print("******* Article number: "+str(i)+" **********")
    num = 0
    vec_sum = 0

    for j in range(len(df["keywords"][i])):
        try:
            vec_temp = vecs.word_vec(df["keywords"][i][j], use_norm=False)
        except Exception as e:
            print(e)
        else:
            num = num + 1
            vec_sum = vec_sum + vec_temp

    try:
        vec_avg = vec_sum/num
        df["vec_sum_avg"][i] = vec_avg
    except Exception as e:
        print(e)
```

Figure 4. 9: Code snippet to calculate vector average and save in dataframe

Chapter 5: Modeling and Evaluation

In this project, the K-means algorithm is used for modeling. This algorithm is described in detail by J. A. Hartigan in 1975. The K-means algorithm aims to divide M points in N dimensions into K-clusters so that the within-cluster sum of squares is minimized. The algorithm requires as input a matrix of M points in N dimensions and a matrix of K initial cluster centers in N dimensions [15].

In this case, M points are drawn in N-dimensional space. The K in K-means is a hyperparameter which determines the number of clusters. In the first step while clustering, it identifies K number of random points in space which is the center of each cluster called centroids. In the next step, the distance of each point in the space is calculated from each centroid and each data point is allocated to a cluster based on the shortest distance. Now centroids are readjusted considering all data points in that cluster. The distance of each data point in the space is recalculated from new centroids, switch the data point allocation to a cluster based on distance and adjust the centroid again. This process is repeated till the point that none of the data points change the cluster and the position of the centroid is fixed.

In this project, the clustering of newspaper articles is done based on the keywords vector average. The vector sum average in the dataframe is converted to a list before passing it to the K-means library method “fit_predict()”. The hyperparameter n_clusters=2 and random_state=0 as shown in Figure 5.1. The clustering output is stored in a new column corresponding to each news article.

3. Modeling ¶

```
In [21]: from sklearn.cluster import KMeans

         X = pd.DataFrame(df['vec_sum_avg'].to_list())
         clusters = KMeans(n_clusters=2, random_state=0).fit_predict(X)
```

```
In [22]: df['cluster'] = clusters
         df.sample(5)
```

Out[22]:

	expanded_url	domain	tag	title	text	summary	keywords	vec_sum_avg	cluster
	s://www.zeit.de/politik/ausland/2021-04/au...	zeit.de	NaN	Lesen Sie zeit.de mit Werbung oder im PUR-Abo....	zeit.de mit Werbung\n\nBesuchen Sie zeit.de wi...	zeit.de mit WerbungBesuchen Sie zeit.de wie ge...	[zeitde, werbung, purabo, tracking, lesen, gew...	[-0.07083344, 0.20493168, -0.09930655, -0.1599...	0
	www.tagesschau.de/ausland/europa/egmr-...	tagesschau.de	NaN	EGMR - Aktuelle Nachrichten	Zerschlagung des Konzerns verletzte Grundrecht...	Zerschlagung des Konzerns verletzte Grundrecht...	[trotzdem, verletzte, russische, unternehmens,...	[-0.09265971, 0.18524776, -0.22347648, -0.4096...	1
	s://m.tagesspiegel.de/politik/anordnung-vo...	tagesspiegel.de	NaN	Anordnung vom Gesundheitsamt: Kinder sollen be...	Ein Brief des Kommunalverbands verängstigt Elt...	In einem Schreiben des Kommunalverbands Region...	[schreiben, anordnung, hannover, isoliert, sie...	[-0.19349097, 0.1434053, -0.14320543, -0.33538...	1
	www.tagesschau.de/faktenfinder/querdenk...	tagesschau.de	NaN	faktenfinder - Aktuelle Nachrichten	Fake News in Deutschland Schneller als die Pol...	Fake News in Deutschland Schneller als die Pol...	[mord, terrorzelle, faktenfinder, patrick, akt...	[-0.19024982, 0.14209354, -0.3217902, -0.26749...	1
	www.tagesschau.de/eilmeldung/corona-de...	tagesschau.de	NaN	Coronavirus - Aktuelle Nachrichten	Nach Urteil aus Karlsruhe Triage-Regelung noch...	Nach Urteil aus Karlsruhe Triage-Regelung noch...	[gesundheitsminister, offenbar, entwurf, aktue...	[-0.12850012, 0.18998758, -0.13915099, -0.3537...	1

Figure 5. 1: Code snippet for K-Means clustering and output values

The clustering is applied to a total of 3760 records. The 2 clusters have 643 records (cluster 0) and 3117 records (cluster 1) as shown in the figure below.

```
In [23]: df0 = df[df.cluster==0]
         df1 = df[df.cluster==1]
         print("Cluster 0 size = "+str(len(df0)))
         print("Cluster 1 size = "+str(len(df1)))

         Cluster 0 size = 643
         Cluster 1 size = 3117
```

Figure 5. 2: Size of 2 clusters after modeling

The K-means algorithm divided the news articles into 2 non-overlapping groups. It used vector sum average for clustering and clustered similar news articles together. Each keyword in a news article is converted to a vector and the average is calculated, these vector averages are compared with each other for similarity. Resulting in the clustering of all articles in such a way that one group of similar articles can be called cluster 0 and another group of similar articles as cluster 1. As part of this project, the following targets are achieved:

- The top five news websites are selected from the dataset.
- Title, text, summary, and keywords are extracted for each news article with the help of natural language processing and Python libraries.

- The keywords are converted to word vectors by the word2vec algorithm and the vector average is calculated
- Unsupervised learning algorithm is used to represent news articles into 2 clusters.

Chapter 8: Conclusion

This project aimed to classify COVID-19 related news articles shared on social media posts. Based on the approach followed in this project, it is concluded that the news articles are divided into clusters where each cluster has group of similar articles. The method followed in this project can be used to classify COVID-19 or any other situation-related news articles spreading on social media platforms. To apply this, a dataset with all news article URLs should be created. The natural language processing methods can be used to extract titles, text, summaries, and keywords using the Newspaper3k library. The word2vec model can be used to convert keywords to vectors and calculate vector averages, which helps to identify similar news articles by comparing vector averages. Finally, one of the available unsupervised machine learning algorithms (e.g.: K-means clustering) can be used to cluster similar news articles.

References

[1] David De Coninck. et al. (2021). Beliefs in Conspiracy Theories and Misinformation About COVID-19: Comparative Perspectives on the Role of Anxiety, Depression and Exposure to and Trust in Information Sources

[2] Rüdiger Wirth and Jochen Hipp. (2000). "CRISP-DM: Towards a Standard Process Model for Data Mining." *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pp. 29–39.

[3] Christoph Schröerab, Felix Kruseb and Jorge Marx Gómezb. (2021). "A Systematic Literature Review on Applying CRISP-DM Process Model", pp. 526-534

[4] Pandas Profiling. https://github.com/ydataai/pandas-profiling (accessed 12th May 2022)

[5] Stephan Richter. "lxml - XML and HTML with Python", https://lxml.de/ (accessed 12th May 2022)

[6] Kenneth Reitz. "Requests: HTTP for Humans", https://docs.python-requests.org/en/latest/ (accessed 12th May 2022)

[7] Xavier Grangier. "Python-Goose - Article Extractor", https://github.com/grangier/python-goose (accessed 12th May 2022)

[8] Lucas Ou-Yang. "Newspaper3k: Article scraping & curation", https://newspaper.readthedocs.io/en/latest/ (accessed 12th May 2022)

[9] Lucas Ou-Yang. "Advanced", https://newspaper.readthedocs.io/en/latest/user_guide/advanced.html#advanced (accessed 12th May 2022)

[10] Radim Řehůřek. "What is Gensim?", https://radimrehurek.com/gensim/intro.html (accessed 12th May 2022)

[11] Tomas Mikolov. et al. (2013). "Efficient Estimation of Word Representations in Vector Space"

[12] Radim Řehůřek. "Why Gensim", https://radimrehurek.com/gensim/index.html (accessed 12th May 2022)

[13] “Wiki word vectors”, https://fasttext.cc/docs/en/pretrained-vectors.html (accessed 12th May 2022)

[14] Piotr Bojanowski et al. (2016). Enriching Word Vectors with Subword Information

[15] J. A. Hartigan and M. A. Wong. (1979). Algorithm AS 136: A k-Means Clustering Algorithm