Emotion detection on ‘WorkFromHome’ related tweets

Course: Natural Language Processing AML_2304_3

Niteesha Balla
Artificial Intelligence andMachine Learning
Lambton College

Toronto, Canada
c0850488@mylambton.ca

Naman Shah
Artificial Intelligence and Machine Learning
Lambton College

Toronto, Canada
c0853267@mylambton.ca

Arjun Chowhan
Artificial Intelligence and Machine Learning
Lambton College

Toronto, Canada
c0850490@mylambton.ca

Abstract— The Work from Home culture has made its place in a Global perspective especially during and after the world having witnessed the effects of the COVID-19 pandemic. In this experiment, we are carrying an analysis on tweets related to the ‘work from home’ keyword and attempting to explore the emotions expressed within the textual data of the tweets using python libraries. This analysis will give us people’s general opinion about working from home and the idea about how this culture will continue in the future days.

Keywords— WFH, Data Fetching, Data Pre-processing, Textual data analysis, Natural Language Processing, Emotion Detection

I. INTRODUCTION

In the past, remote employees have had a bad reputation. Many employers believed their workforce would be too easily distracted at home, where their managers couldn’t keep an eye on their direct reports [1]. Today, remote work has made its way to the society due to its mandatory need in the pandemic and has proven efficient enough to be kept ongoing as the new normal in some organizations even after the reduced effects of the virus outbreak. We were fascinated to dig into the banter related to work from home on twitter, and so decided to explore what emotion do the tweets express to get a notion about the feelings of people regarding this culture. Also, natural language processing techniques and inbuilt packages in Python boosted our interest of moving forward with this experimental analysis and text-based emotion detection.

II. DATASET DESCRIPTION

Our dataset consists of about 10,362 extracted tweets using the tweepy library in python and Twitter Developer Account API access tokens. Tweepy is a Python package that makes it simple to use the Twitter API. The Twitter Developer Account Elevated Access provides us with API Key’s to analyze, learn from, and interact with Tweets, Direct Messages, and users. The extracted tweets are stored in a CSV file using Pandas library.

3 attributes have been extracted as columns:

i. Date and time of tweet creation
ii. User Id (id of user who tweeted)
iii. Text content of the tweet

The main attribute we will be focusing on for this analysis is the Text content of the tweets. The tweets are a recent pick of data from dates July 16 to July 26, 2022.

III. EXPERIMENT

We are using Python Jupiter Notebook for our project analysis and showcasing the results. It is a free tool and easily helps us edit, execute, and visualize every statement output of Python code, and present it in an explanatory manner.

Firstly we, extracted the tweets stored into CSV file and used Panda’s python data frame to display it. Below is the preview of our initial data.

```
wfh_df.head()
```

	created_at	user_id	text
0	2022-07-26 20:03:14+00:00	16117160	Saw this @WSJ article on remote work coinciden...
1	2022-07-26 20:01:20+00:00	4440814155	Much of the discussion surrounding the office ...
2	2022-07-26 19:57:28+00:00	2176790760	New Remote Customer Support Job! Stock&Buy...
3	2022-07-26 19:56:55+00:00	229540362	The most recommended #WFH tips from remote emp...
4	2022-07-26 19:47:38+00:00	2176790760	New Remote Management and Finance Job! Cloudli...

Next, we carried out preprocessing(cleaning) on our ‘text’ data. The steps involved in the cleaning process are as follows:

1. Removing Punctuations [,*^!”/]
2. Removing Numbers [193234]
3. Removing HTML tags [<head>,
]
4. Removing URL functions
5. Removing Accented characters
6. Removing Extra Whitespaces

7. Removing duplicate records.

Pre-processing

```
def remove_punctuation_func(text):
    return re.sub(r'[^a-zA-Z0-9]', ' ', text)

def remove_numbers(text):
    return re.sub(r'[0-9]', ' ', text)

def remove_html_tags_func(text):
    return BeautifulSoup(text, 'html.parser').get_text()

def remove_url_func(text):
    return re.sub(r'https?://\S+|www\.\S+', '', text)

def remove_accented_chars_func(text):
    return unicodedata.normalize('NFKD', text).encode('ascii', 'ignore').decode('utf-8', 'ignore')

def remove_extra_whitespaces_func(text):
    return re.sub(r'^\s*|\s\s*', ' ', text).strip()
```

```
def pre_processing(dataset):
    #drop duplicates
    dataset.drop_duplicates(inplace=True)

    #convert to lowercase
    dataset["text"] = dataset["text"].str.lower()
```

Text after preprocessing: -

```
wfh_df.head()
```

	created_at	user_id	text
0	2022-07-26 20:03:14+00:00	16117160	saw this wsj article on remote work coincident...
1	2022-07-26 20:01:20+00:00	4440814155	much of the discussion surrounding the office ...
2	2022-07-26 19:57:28+00:00	2176790760	new remote customer support job stock amp buy ...
3	2022-07-26 19:56:55+00:00	229540362	the most recommended wfh tips from remote empl...
4	2022-07-26 19:47:38+00:00	2176790760	new remote management and finance job cloudlin...

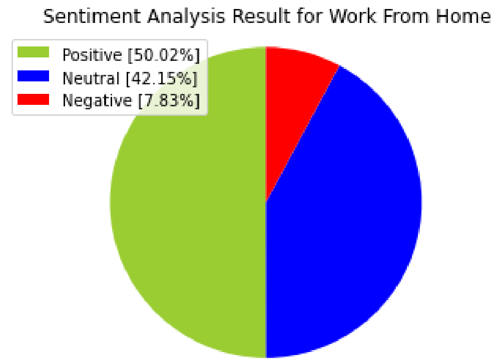
All the above steps ensure that there are no unwanted characters in the text data, that happen to be noise in the textual analysis and understanding the sentiments. We use the re (regular expression) package for removal of punctuations, numbers, accented characters, and whitespaces. We used beautiful soup library to remove HTML tag content from the text.

Below is the preview of processed text obtained after cleaning:

text
saw this wsj article on remote work coincident...
much of the discussion surrounding the office ...
new remote customer support job stock amp buy ...
the most recommended wfh tips from remote empl...
new remote management and finance job cloudlin...

Further, we began our analysis by using an inbuilt Python library called TextBlob to understand the Sentiment polarity for each of the tweets. We fed all the tweets into the Sentiment Analyzer method of this library and obtained the sentiment scores of the tweets, categorizing them into Positive, Negative and Neutral tweets.

The sentiment polarity scores determine the category of the tweet text. A negative score closer to -1 indicates the negative sentiment of a tweet. A positive score closer to 1 indicates the positive sentiment of tweet, where as a score near to 0 indicates a neutral sentiment. Below pie diagram shows how our tweets categorized:



Here, we find that 50 % tweets in our data express a positive sentiment, 42 % being neutral and about 8 % tweets have a negative sentiment.

We have also used Vader sentiment Analysis to understand the polarity of the tweets. The lexicon- and rule-based sentiment analysis tool VADER (Valence Aware Dictionary and Sentiment Reasoner) is customized precisely to sentiments expressed on social media. Vader makes use of a variety of sentiment lexicon is a collection of lexical elements that are often classified as either positive or negative depending on their semantic orientation. Vader not only informs us of the positivity and negativity scores, but also of the sentimentality of each score.

The below image shows a preview of the results generated by TextBlob and Vader sentiment analyzer:

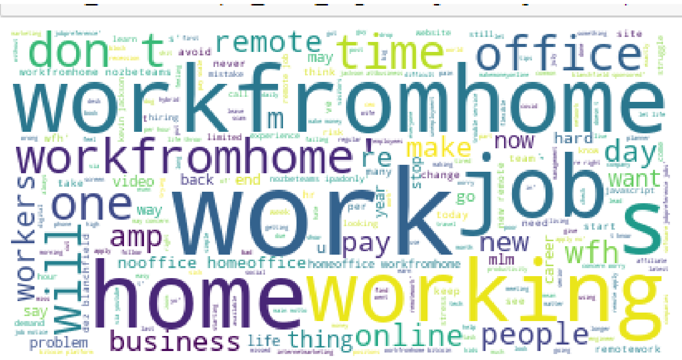
text	polarity	subjectivity	sentiment	neg	neu	pos	compound
saw this wsj article on remote work coincident...	-0.050000	0.162500	neutral	0.0	1.000	0.000	0.0000
much of the discussion surrounding the office ...	0.200000	0.200000	positive	0.0	0.890	0.110	0.3818
new remote customer support job stock amp buy ...	0.018182	0.327273	positive	0.0	0.779	0.221	0.6597
the most recommended wfh tips from remote empl...	0.200000	0.350000	positive	0.0	0.870	0.130	0.2716
new remote management and finance job cloudlin...	-0.021212	0.284848	neutral	0.0	1.000	0.000	0.0000

The 'polarity' is an output of TextBlob, whereas the attributes 'neu', 'pos','neg' and 'compound' are Output of Vader algorithm.

Further, we attempted to visualize the significant words within positive and negative tweets. We used the WordCloud library in Python. A word cloud, commonly referred to as a tag cloud, is an image of words. Popular words and phrases are highlighted using cloud creators depending on their frequency and importance. They give you immediate, straightforward visual insights that can inspire more thorough analysis.

Above image shows a word cloud for positively categorized tweets where some words are ‘support’, ‘focused’, ‘inspire’,’no office’ are seen.

Below ` image shows a word cloud for positively categorized tweets where some words are ‘hard’, ‘stop’, ‘change’, ‘problem’ are seen.



Emotion Detection:

Finally, for our end goal being emotion detection, we made use of an emotion dataset which is inbuilt within the nlp package of python. The six primary emotions included in the Emotion dataset are anger, fear, joy, love, sadness, and surprise.

Below are some tweets from the dataset.

```
tweets[10], labels[10]
```

```
('i feel like i have to make the suffering i m seeing mean something',
 'sadness')
```

```
tweets[:10]
```

```
['i didnt feel humiliated',
 'i can go from feeling so hopeless to so damned hopeful just from being around someone who cares and is awake',
 'im grabbing a minute to post i feel greedy wrong',
 'i am ever feeling nostalgic about the fireplace i will know that it is still on the property',
 'i am feeling grouchy',
 'ive been feeling a little burdened lately wasnt sure why that was',
 'ive been taking or milligrams or times recommended amount and ive fallen asleep a lot faster but i also feel like so funny',
 'i feel as confused about life as a teenager or as jaded as a year old man',
 'i have been with petronas for years i feel that petronas has performed well and made a huge profit',
 'i feel romantic too']
```

We created training and test dataframes and then used neural network models in keras and tensorflow libraries to implement the model training on our data.

We tokenize the tweets using tokenizer function in TensorFlow. Tokenization splits the sentence into separate words and appends all the words into a list.

Tokenizing the wfh tweets

```
from tensorflow.keras.preprocessing.text import Tokenizer
```

```
wfh_tweets = list(wfh_df['text'])
tokenizer.fit_on_texts(wfh_tweets)
```

```
65
```

Padding and Truncating

```
from tensorflow.keras.preprocessing.sequence import pad_sequences
```

```
wfh_tokens = get_sequences(tokenizer, wfh_tweets)
wfh_tokens[10]
```

```
array([  30,   18,   46,  120,   18,   34, 1302, 7413, 3214,  333, 2102,
         64,   49,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
          0])
```

```
len(wfh_tokens[10])
```

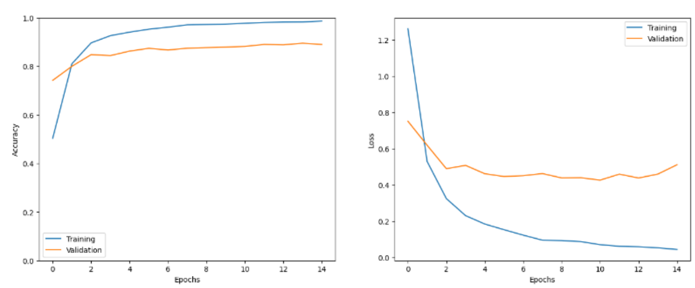
```
100
```

Our neural network model is a sequential network with 1 input layer of 50 neurons, 2 bidirectional hidden layers that use LSTM (Long-short-term memory) of 20 neurons each and 1 output layer of 6 neurons that uses SoftMax activation function. We are using Sparse_categorical_entropy as the loss function and Adam optimizer to compile the model.

After training the model with 15 epochs (iterations), we observed the accuracy value reaching 97 %. Hence, our neural network model can be called ready to make fair predictions for the different emotions being ‘joy, ‘sadness’, ‘fear’, ‘anger’, ‘love’, and ‘surprise’.

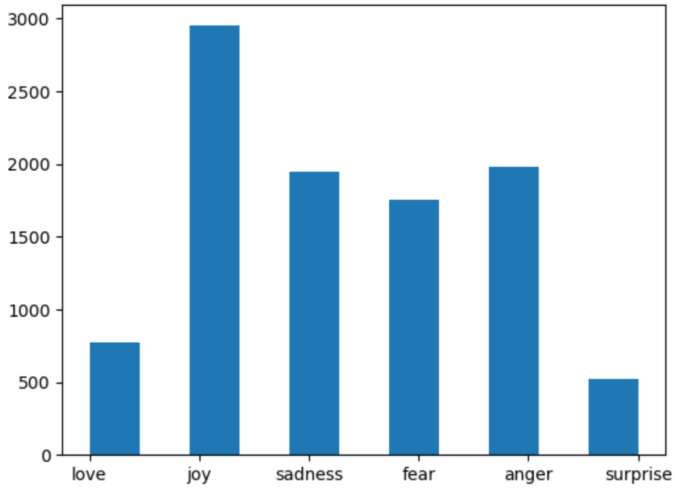
```
Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_6 (Embedding)     (None, 60, 16)            160000

 bidirectional_12 (Bidirecti  (None, 60, 40)           5920
 onal)

 bidirectional_13 (Bidirecti  (None, 40)               9760
 onal)

 dense_6 (Dense)             (None, 6)                 246

=================================================================
Total params: 175,926
Trainable params: 175,926
Non-trainable params: 0
_________________________________________________________________
```

Training and validation graph.



Following histogram image shows the results of our model for emotion detection:

The graph shoes the count of tweets classified into 6 emotions. We observe that most tweets express the emotion of joy, but there are considerable number of tweets that also express sadness fear and anger.



IV. Results

Through our analysis, we found out that majority of the tweets express the emotion of ‘Joy’. Although there are considerable number of tweets that also express ‘Sadness’.

Here is the count of the tweets with their respective emotions.

```
print("love:", wfh_labels.count('love'))
print("joy:", wfh_labels.count('joy'))
print("sadness:", wfh_labels.count('sadness'))
print("fear:", wfh_labels.count('fear'))
print("anger:", wfh_labels.count('anger'))
print("surprise:", wfh_labels.count('surprise'))
```

```
love: 772
joy: 2949
sadness: 1942
fear: 1753
anger: 1978
surprise: 523
```

Below are some random tweets with their emotions. The model predicts almost correct for some tweets.

```
Tweet: newly developed system generating life changing wealth currently looking for additional beta testers use this link
Predicted Emotion: love


Tweet: change management is an often overlooked skill but critical to your organization s success
Predicted Emotion: sadness


Tweet: if you have any skill then you can offer it online to make money here fiverr
Predicted Emotion: joy


Tweet: tuesday s tip of the week prevent your printer ink cartridges from drying out here s how remove your ink cart
Predicted Emotion: fear


Tweet: turn your cellphone into your personal atm text info to opportunity income workfromhome
Predicted Emotion: sadness
```

V. CONCLUSION

Our analysis may not be accurate enough for all the predicted emotions, but we can get a notion of people having more positive opinion about the work from home culture. However, for people who like to socialize during work, their lesser interaction with colleagues in-person can bring their feelings of sadness and frustration while working.

VI. REFERECENES

[1] Peek, S. (2018, December 27). *Current and future trends in remote work - businessnewsdaily.com*. Business News Daily. https://www.businessnewsdaily.com/8156-future-of-remote-work.html

[2] Analytics Vidhya. 2021. *Emotion classification on Twitter Data Using Transformers*. [online] Available at: <https://www.analyticsvidhya.com/blog/2021/01/emotion-classification-on-twitter-data-using-transformers/> [Accessed 13 August 2022].

[3]*Twitter-emotion-Recognition/Twitter emotion recognition using RNN.ipynb at main · katoch99/Twitter-emotion-Recognition*. (n.d.). GitHub. https://github.com/katoch99/Twitter-Emotion-Recognition/blob/main/Twitter%20Emotion%20Recognition%20using%20RNN.ipynb

[4] *NLP - Text pre-processing II (Tokenization and stop words)*. (2021, May 25). Michael Fuchs Python. https://michael-fuchs-python.netlify.app/2021/05/25/nlp-text-pre-processing-ii-tokenization-and-stop-words/

[5] *Tweet emotion detection | Kaggle*. (n.d.). Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/competitions/tweet-emotion-detection/code