Design Project - Entity Based Sentiment Analysis On Twitter Data

November 27, 2022

0.1 Imports

[9]:
```
#Data Management
import pandas as pd
import numpy as np
np.random.seed(0)

#TextBlob Features
from textblob import TextBlob

#Plotting
import matplotlib.pyplot as plt

#SciKit-Learn
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
from sklearn.metrics import classification_report

#Save Data
import pickle

#nltk
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from nltk.stem import WordNetLemmatizer

#Tensorflow / Keras
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow import keras

#Visualisation
import seaborn as sns

#Tokenize
from collections import Counter
```

```
#Warnings Filter
import warnings
warnings.filterwarnings("ignore")
```

0.2 Loading Data

[2]:
```
#Train Data
path = "datasets/twitter_training.csv"
train_df = pd.read_csv(path, names=["Tweet_ID", "Entity", "Sentiment",␣
 ↪"Tweet_Content"])

#Test Data
test_path = "datasets/twitter_validation.csv"
test_df = pd.read_csv(test_path, names=["Tweet_ID", "Entity", "Sentiment",␣
 ↪"Tweet_Content"])
```

[4]:
```
# #Adding a single input
# new_inp_list = ["12345", "Amazon", "Positive", "Amazon is a really cool␣
 ↪company. I think it's great"]

# #Setting it at the begining of the dataset
# test_df.iloc[0] = new_inp_list
```

0.3 Exploraotry Data Analysis

[3]:
```
train_df
```

[3]:
```
       Tweet_ID       Entity Sentiment  \
0          2401  Borderlands  Positive
1          2401  Borderlands  Positive
2          2401  Borderlands  Positive
3          2401  Borderlands  Positive
4          2401  Borderlands  Positive
…           …           …         …
74677      9200       Nvidia  Positive
74678      9200       Nvidia  Positive
74679      9200       Nvidia  Positive
74680      9200       Nvidia  Positive
74681      9200       Nvidia  Positive

                                         Tweet_Content
0      im getting on borderlands and i will murder yo…
1      I am coming to the borders and I will kill you…
2      im getting on borderlands and i will kill you …
3      im coming on borderlands and i will murder you…
4      im getting on borderlands 2 and i will murder …
…                                                    …
```

```
74677  Just realized that the Windows partition of my…
74678  Just realized that my Mac window partition is …
74679  Just realized the windows partition of my Mac …
74680  Just realized between the windows partition of…
74681  Just like the windows partition of my Mac is l…

[74682 rows x 4 columns]
```

[4]:
```
test_df
```

[4]:
```
     Tweet_ID              Entity   Sentiment  \
0        3364            Facebook  Irrelevant
1         352              Amazon     Neutral
2        8312           Microsoft    Negative
3        4371               CS-GO    Negative
4        4433              Google     Neutral
..        …                   …         …
995      4891  GrandTheftAuto(GTA)  Irrelevant
996      4359               CS-GO  Irrelevant
997      2652         Borderlands    Positive
998      8069           Microsoft    Positive
999      6960     johnson&johnson     Neutral

                                          Tweet_Content
0    I mentioned on Facebook that I was struggling …
1    BBC News - Amazon boss Jeff Bezos rejects clai…
2    @Microsoft Why do I pay for WORD when it funct…
3    CSGO matchmaking is so full of closet hacking,…
4    Now the President is slapping Americans in the…
..                                                 …
995   Toronto is the arts and culture capital of …
996  tHIS IS ACTUALLY A GOOD MOVE TOT BRING MORE VI…
997  Today sucked so it’s time to drink wine n play…
998  Bought a fraction of Microsoft today. Small wins.
999  Johnson & Johnson to stop selling talc baby po…

[1000 rows x 4 columns]
```

[5]:
```
train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 74682 entries, 0 to 74681
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Tweet_ID       74682 non-null  int64
 1   Entity         74682 non-null  object
 2   Sentiment      74682 non-null  object
```

```
 3   Tweet_Content  73996 non-null  object
dtypes: int64(1), object(3)
memory usage: 2.3+ MB
```

[6]:
```
test_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 4 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   Tweet_ID       1000 non-null   int64 
 1   Entity         1000 non-null   object
 2   Sentiment      1000 non-null   object
 3   Tweet_Content  1000 non-null   object
dtypes: int64(1), object(3)
memory usage: 31.4+ KB
```

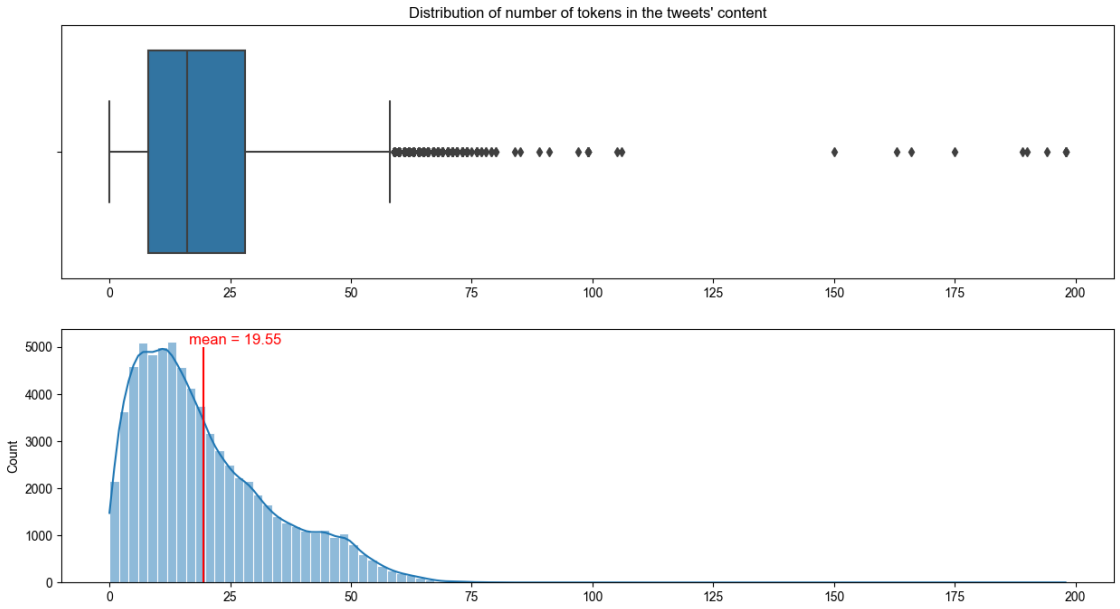
[7]:
```
#drop nulls
train_df = train_df.dropna()
test_df = test_df.dropna()
```

[10]:
```
#drop duplicates
train_df.drop_duplicates(inplace=True)
```

[11]:
```
#analysing tweets
content = train_df['Tweet_Content']

content_lens = [len(i.split()) for i in content.values] #length of tweets
mean_len = np.mean(content_lens)

fig, axes = plt.subplots(2, 1, figsize=(15, 8))
axes[0].set_title('Distribution of number of tokens in the tweets\' content')
sns.set_theme(style="whitegrid", palette='tab10')
sns.boxplot(content_lens, ax=axes[0])
sns.histplot(content_lens, bins=100, kde=True, ax=axes[1])
axes[1].vlines(mean_len, 0, 5000, color = 'r')
plt.annotate("mean = " +  str(round(mean_len, 2)), xy=(mean_len, 5000),␣
 ↪xytext=(mean_len - 3, 5050), color='r')
plt.show()
```



[12]:
```
train_df[train_df['Sentiment'] == 'Positive']
```

[12]:

	Tweet_ID	Entity	Sentiment	Tweet_Content
0	2401	Borderlands	Positive	im getting on borderlands and i will murder yo…
1	2401	Borderlands	Positive	I am coming to the borders and I will kill you…
2	2401	Borderlands	Positive	im getting on borderlands and i will kill you …
3	2401	Borderlands	Positive	im coming on borderlands and i will murder you…
4	2401	Borderlands	Positive	im getting on borderlands 2 and i will murder …
…	…	…	…	…
74677	9200	Nvidia	Positive	Just realized that the Windows partition of my…
74678	9200	Nvidia	Positive	Just realized that my Mac window partition is …
74679	9200	Nvidia	Positive	Just realized the windows partition of my Mac …
74680	9200	Nvidia	Positive	Just realized between the windows partition of…
74681	9200	Nvidia	Positive	Just like the windows partition of my Mac is l…

```
[19713 rows x 4 columns]
```

[13]:
```
train_df[train_df['Sentiment'] == 'Negative']
```

[13]:
```
       Tweet_ID       Entity Sentiment  \
24         2405  Borderlands  Negative
25         2405  Borderlands  Negative
27         2405  Borderlands  Negative
28         2405  Borderlands  Negative
29         2405  Borderlands  Negative
…           …           …       …
74665      9198       Nvidia  Negative
74666      9198       Nvidia  Negative
74667      9198       Nvidia  Negative
74668      9198       Nvidia  Negative
74669      9198       Nvidia  Negative

                                            Tweet_Content
24     the biggest dissappoinment in my life came out…
25     The biggest disappointment of my life came a y…
27     the biggest dissappoinment in my life coming o…
28     For the biggest male dissappoinment in my life…
29     the biggest dissappoinment in my life came bac…
…                                                    …
74665         Nvidia really delayed the 3070 by 2 weeks.
74666                  Nvidia did delay by 3070 2 weeks.
74667      Nvidia really delayed the 3070 several weeks.
74668  Nvidia really only delayed the 3070 2 flight w…
74669            Nvidia really delayed the next 2 weeks.

[21698 rows x 4 columns]
```

[14]:
```
train_df[train_df['Sentiment'] == 'Neutral']
```

[14]:
```
       Tweet_ID       Entity Sentiment  \
12         2403  Borderlands   Neutral
13         2403  Borderlands   Neutral
14         2403  Borderlands   Neutral
15         2403  Borderlands   Neutral
16         2403  Borderlands   Neutral
…           …           …       …
74659      9197       Nvidia   Neutral
74660      9197       Nvidia   Neutral
74661      9197       Nvidia   Neutral
74662      9197       Nvidia   Neutral
74663      9197       Nvidia   Neutral
```

```
                                                Tweet_Content
12     Rock-Hard La Varlope, RARE & POWERFUL, HANDSOM…
13     Rock-Hard La Varlope, RARE & POWERFUL, HANDSOM…
14     Rock-Hard La Varlope, RARE & POWERFUL, HANDSOM…
15     Rock-Hard La Vita, RARE BUT POWERFUL, HANDSOME…
16     Live Rock - Hard music La la Varlope, RARE & t…
…                                                    …
74659  Nvidia plans to release its 2017 "Crypto Craze…
74660  Nvidia does not want to give up its "cryptoins…
74661  Nvidia doesn't intend to give away its 2017 ad…
74662  Nvidia therefore doesn ' t want to give up its…
74663  is doesn't should I give up its password 'cryp…

[17708 rows x 4 columns]
```

[15]:
```
train_df[train_df['Sentiment'] == 'Irrelevant']
```

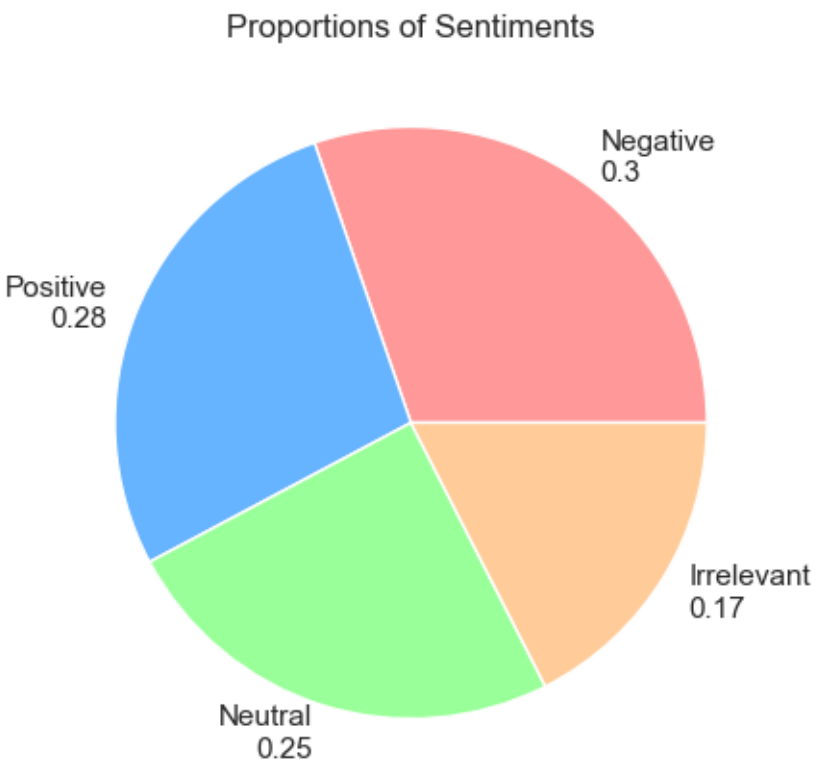
[15]:
```
       Tweet_ID       Entity   Sentiment  \
102        2418  Borderlands  Irrelevant
103        2418  Borderlands  Irrelevant
104        2418  Borderlands  Irrelevant
105        2418  Borderlands  Irrelevant
106        2418  Borderlands  Irrelevant
…           …            …           …
74035      9085       Nvidia  Irrelevant
74036      9085       Nvidia  Irrelevant
74037      9085       Nvidia  Irrelevant
74038      9085       Nvidia  Irrelevant
74039      9085       Nvidia  Irrelevant

                                                Tweet_Content
102    Appreciate the (sonic) concepts / praxis Valen…
103    Appreciate the (sound) concepts / practices th…
104    Evaluate the (sound) concepts / concepts of Va…
105    Appreciate the (sonic) concepts / praxis Valen…
106    Appreciate by the ( sonic ) electronic concept…
…                                                    …
74035  This is all based on last quarter's earnings. …
74036  Let's see how well they handle the next one wh…
74037  Good on them. This stuff all based on earnings…
74038  9 Good idea for them. This is all based on ear…
74039  goes for them. This is all based on earnings f…

[12537 rows x 4 columns]
```

[16]:
```
#analysing sentiments
sentiment_ratios = train_df['Sentiment'].value_counts()
plt.figure(figsize=(5, 5))
plt.pie(sentiment_ratios, labels=[f'{idx}\n{round(sentiment_ratios[idx]/
 ↪len(train_df), 2)}' for idx in sentiment_ratios.index], colors =␣
 ↪['#ff9999','#66b3ff','#99ff99','#ffcc99'])
plt.title('Proportions of Sentiments')
plt.show()

print(sentiment_ratios)
```




```
Negative      21698
Positive      19713
Neutral       17708
Irrelevant    12537
Name: Sentiment, dtype: int64
```

[17]:
```
#analysing entities
print("\nEntities:", end =" ")
print(len(train_df['Entity'].unique()), end = "\n\n")
print(train_df['Entity'].value_counts())
```

```
Entities: 32

TomClancysRainbowSix                  2328
Verizon                               2319
MaddenNFL                             2315
CallOfDuty                            2314
Microsoft                             2304
WorldOfCraft                          2300
NBA2K                                 2299
LeagueOfLegends                       2296
TomClancysGhostRecon                  2291
Facebook                              2289
ApexLegends                           2278
johnson&johnson                       2257
Battlefield                           2255
Amazon                                2249
CallOfDutyBlackopsColdWar             2242
FIFA                                  2238
Dota2                                 2225
Overwatch                             2220
Hearthstone                           2219
HomeDepot                             2216
GrandTheftAuto(GTA)                   2208
Borderlands                           2206
Xbox(Xseries)                         2201
Google                                2199
Nvidia                                2198
CS-GO                                 2195
PlayStation5(PS5)                     2183
Fortnite                              2176
Cyberpunk2077                         2175
AssassinsCreed                        2156
RedDeadRedemption(RDR)                2155
PlayerUnknownsBattlegrounds(PUBG)     2150
Name: Entity, dtype: int64
```

0.4 Pre-processing

[18]:
```
df = train_df
```

[19]:
```
#encoding entity information
onehot = pd.get_dummies(df["Entity"], prefix="Entity")

#adding to dataframe
df = df.join(onehot)
```

[20]:
```
#for test data
onehot = pd.get_dummies(test_df["Entity"], prefix="Entity")

test_df = test_df.join(onehot)
```

[21]:
```
#enriching with textblob
def tb_enrich(ls):
    tb_polarity = []
    tb_subject = []

    for tweet in ls:
        tb_polarity.append(TextBlob(tweet).sentiment[0])
        tb_subject.append(TextBlob(tweet).sentiment[1])

    return tb_polarity, tb_subject
```

[22]:
```
#train data
df["Polarity"], df["Subjectivity"] = tb_enrich(list(df["Tweet_Content"]))
```

[23]:
```
#test data
test_df["Polarity"], test_df["Subjectivity"] =␣
 ↪tb_enrich(list(test_df["Tweet_Content"]))
```

[24]:
```
#encodes senitment strings to numbers and vice versa
class_to_index = {"Neutral":0, "Irrelevant":1, "Negative":2, "Positive": 3}
index_to_class = dict((v,k) for k, v in class_to_index.items())

#applying for each entry in the dataset
names_to_ids = lambda n: np.array([class_to_index.get(x) for x in n])
ids_to_names = lambda n: np.array([index_to_class.get(x) for x in n])
```

[25]:
```
#train
df["Sentiment"] = names_to_ids(df["Sentiment"])
y = df["Sentiment"]
```

[26]:
```
#test
test_df["Sentiment"] = names_to_ids(test_df["Sentiment"])
y_test = test_df["Sentiment"]
```

[27]:
```
lemmatiser = WordNetLemmatizer() #word root form
stop_english = Counter(stopwords.words()) #for iterating over stop words
```

[28]:
```
#converts words to root form and removes stop words
def remove_stopwords(ls):
    ls = [lemmatiser.lemmatize(word) for word in ls if word not in␣
 ↪(stop_english) and (word.isalpha())]
    ls = " ".join(ls)
```

```
    return ls
```

[29]:
```
#tokenizes text to words
df["Tweet_Content_Split"] = df["Tweet_Content"].apply(word_tokenize)

#removes stopwords
df["Tweet_Content_Split"] = df["Tweet_Content_Split"].apply(remove_stopwords)
```

[30]:
```
#test data
test_df["Tweet_Content_Split"] = test_df["Tweet_Content"].apply(word_tokenize)

test_df["Tweet_Content_Split"] = test_df["Tweet_Content_Split"].
 ↪apply(remove_stopwords)
```

[31]:
```
#Vectorization

#setting for 1000 most frequent words
tokeniser = Tokenizer(num_words=1000, lower=True)

#top 1000 words present in text
tokeniser.fit_on_texts(df["Tweet_Content_Split"])

#converts the tokens to number features
tweet_tokens = tokeniser.texts_to_matrix(list(df["Tweet_Content_Split"]))
```

[17]:
```
# # saving the tokenizer
# with open('saved_tokenizer.pkl', 'wb') as handle:
#     pickle.dump(tokeniser, handle, protocol=pickle.HIGHEST_PROTOCOL)
```

[19]:
```
# # loading the tokenizer
# with open('saved_tokenizer.pkl', 'rb') as handle:
#     tokeniser = pickle.load(handle)
```

[32]:
```
#test data
tweet_tokens_test = tokeniser.
 ↪texts_to_matrix(list(test_df["Tweet_Content_Split"]))
```

[33]:
```
tweet_tokens
```

[33]:
```
array([[0., 0., 0., …, 0., 0., 0.],
       [0., 1., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       …,
       [0., 1., 0., …, 0., 0., 0.],
       [0., 1., 0., …, 0., 0., 0.],
       [0., 1., 0., …, 0., 0., 0.]])
```

[34]:
```
tweet_tokens_test
```

[34]:
```
array([[0., 1., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 1., 0., …, 0., 0., 0.],
       …,
       [0., 1., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.],
       [0., 0., 0., …, 0., 0., 0.]])
```

[35]:
```
#reset df index
df.reset_index(drop=True, inplace=True)
test_df.reset_index(drop=True, inplace=True)
```

[36]:
```
#joining encoded data to existing df
full_df = pd.concat([df, pd.DataFrame(tweet_tokens)], sort=False, axis=1)
```

[37]:
```
#test data
full_test_df = pd.concat([test_df, pd.DataFrame(tweet_tokens_test)],␣
 ↪sort=False, axis=1)
```

[38]:
```
#dropping raw data
full_df = full_df.drop(["Sentiment", "Tweet_ID", "Tweet_Content",␣
 ↪"Tweet_Content_Split", "Entity"], axis=1)
```

[39]:
```
#test data
full_test_df = full_test_df.drop(["Sentiment", "Tweet_ID", "Tweet_Content",␣
 ↪"Tweet_Content_Split", "Entity"], axis=1)
```

[40]:
```
full_df
```

[40]:

	Entity_Amazon	Entity_ApexLegends	Entity_AssassinsCreed	\
0	0	0	0	
1	0	0	0	
2	0	0	0	
3	0	0	0	
4	0	0	0	
…	…	…	…	
71651	0	0	0	
71652	0	0	0	
71653	0	0	0	
71654	0	0	0	
71655	0	0	0	

	Entity_Battlefield	Entity_Borderlands	Entity_CS-GO	\
0	0	1	0	
1	0	1	0	

```
2                      0                    1              0
3                      0                    1              0
4                      0                    1              0
…                    …                    …            …
71651                  0                    0              0
71652                  0                    0              0
71653                  0                    0              0
71654                  0                    0              0
71655                  0                    0              0

       Entity_CallOfDuty  Entity_CallOfDutyBlackopsColdWar  \
0                      0                                  0
1                      0                                  0
2                      0                                  0
3                      0                                  0
4                      0                                  0
…                    …                                  …
71651                  0                                  0
71652                  0                                  0
71653                  0                                  0
71654                  0                                  0
71655                  0                                  0

       Entity_Cyberpunk2077  Entity_Dota2  …  990  991  992  993  994  995  \
0                         0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
1                         0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
2                         0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
3                         0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
4                         0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
…                       …           …  …  …  …  …  …  …  …
71651                     0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
71652                     0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
71653                     0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
71654                     0             0  …  0.0  0.0  0.0  0.0  0.0  0.0
71655                     0             0  …  0.0  0.0  0.0  0.0  0.0  0.0

       996  997  998  999
0      0.0  0.0  0.0  0.0
1      0.0  0.0  0.0  0.0
2      0.0  0.0  0.0  0.0
3      0.0  0.0  0.0  0.0
4      0.0  0.0  0.0  0.0
…    …  …  …  …
71651  0.0  0.0  0.0  0.0
71652  0.0  0.0  0.0  0.0
71653  0.0  0.0  0.0  0.0
71654  0.0  0.0  0.0  0.0
```

```
71655  0.0  0.0  0.0  0.0

[71656 rows x 1034 columns]
```

[41]:
```
full_test_df
```

[41]:
```
      Entity_Amazon  Entity_ApexLegends  Entity_AssassinsCreed  \
0                 0                   0                      0
1                 1                   0                      0
2                 0                   0                      0
3                 0                   0                      0
4                 0                   0                      0
..              …                   …                      …
995               0                   0                      0
996               0                   0                      0
997               0                   0                      0
998               0                   0                      0
999               0                   0                      0

     Entity_Battlefield  Entity_Borderlands  Entity_CS-GO  Entity_CallOfDuty  \
0                     0                   0             0                  0
1                     0                   0             0                  0
2                     0                   0             0                  0
3                     0                   0             1                  0
4                     0                   0             0                  0
..                  …                   …             …                  …
995                   0                   0             0                  0
996                   0                   0             1                  0
997                   0                   1             0                  0
998                   0                   0             0                  0
999                   0                   0             0                  0

     Entity_CallOfDutyBlackopsColdWar  Entity_Cyberpunk2077  Entity_Dota2  \
0                                   0                     0             0
1                                   0                     0             0
2                                   0                     0             0
3                                   0                     0             0
4                                   0                     0             0
..                                …                     …             …
995                                 0                     0             0
996                                 0                     0             0
997                                 0                     0             0
998                                 0                     0             0
999                                 0                     0             0

     …  990  991  992  993  994  995  996  997  998  999
0    …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
```

```
1    …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
2    …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
3    …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
4    …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
..   …  …  …  …  …  …  …  …  …  …  …
995  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
996  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
997  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
998  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
999  …  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0

[1000 rows x 1034 columns]
```

[73]:
```
# #save prepared data for the future

# with open('full_df.pkl', 'wb') as f:
#     pickle.dump(full_df, f)
# with open('full_test_df.pkl', 'wb') as f:
#     pickle.dump(full_test_df, f)

# with open('y.pkl', 'wb') as f:
#     pickle.dump(y, f)
# with open('y_test.pkl', 'wb') as f:
#     pickle.dump(y_test, f)
```

[2]:
```
# #load saved data

# with open('full_df.pkl', 'rb') as f:
#     full_df = pickle.load(f)
# with open('full_test_df.pkl', 'rb') as f:
#     full_test_df = pickle.load(f)

# with open('y.pkl', 'rb') as f:
#     y = pickle.load(f)
# with open('y_test.pkl', 'rb') as f:
#     y_test = pickle.load(f)
```

0.5 Training the Model

[42]:
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Dense(36, input_dim=1034, activation='relu'),
    tf.keras.layers.Dense(20, activation='relu'),
    tf.keras.layers.Dense(4, activation='sigmoid')
])
model.compile(
     loss='sparse_categorical_crossentropy',
     optimizer='adam',
```

```
    metrics=['accuracy']
)
```

[43]:
```
h = model.fit(
    full_df, y,
    validation_data=(full_test_df, y_test),
    epochs=30,
    callbacks=[tf.keras.callbacks.EarlyStopping(monitor='accuracy',␣
 ↪patience=5)]
)
```

```
Epoch 1/30
2240/2240 [==============================] - 6s 3ms/step - loss: 0.9771 -
accuracy: 0.5888 - val_loss: 0.7227 - val_accuracy: 0.7260
Epoch 2/30
2240/2240 [==============================] - 5s 2ms/step - loss: 0.7657 -
accuracy: 0.6926 - val_loss: 0.5588 - val_accuracy: 0.8040
Epoch 3/30
2240/2240 [==============================] - 5s 2ms/step - loss: 0.6170 -
accuracy: 0.7598 - val_loss: 0.4570 - val_accuracy: 0.8500
Epoch 4/30
2240/2240 [==============================] - 5s 2ms/step - loss: 0.5122 -
accuracy: 0.8043 - val_loss: 0.3875 - val_accuracy: 0.8820
Epoch 5/30
2240/2240 [==============================] - 5s 2ms/step - loss: 0.4416 -
accuracy: 0.8324 - val_loss: 0.3510 - val_accuracy: 0.9000
Epoch 6/30
2240/2240 [==============================] - 5s 2ms/step - loss: 0.3897 -
accuracy: 0.8523 - val_loss: 0.3315 - val_accuracy: 0.8980
Epoch 7/30
2240/2240 [==============================] - 5s 2ms/step - loss: 0.3545 -
accuracy: 0.8666 - val_loss: 0.3309 - val_accuracy: 0.8980
Epoch 8/30
2240/2240 [==============================] - 6s 3ms/step - loss: 0.3264 -
accuracy: 0.8765 - val_loss: 0.3137 - val_accuracy: 0.9020
Epoch 9/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.3043 -
accuracy: 0.8852 - val_loss: 0.3291 - val_accuracy: 0.9060
Epoch 10/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2878 -
accuracy: 0.8911 - val_loss: 0.3192 - val_accuracy: 0.9060
Epoch 11/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2707 -
accuracy: 0.8976 - val_loss: 0.3119 - val_accuracy: 0.9090
Epoch 12/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2588 -
accuracy: 0.9015 - val_loss: 0.3248 - val_accuracy: 0.9110
Epoch 13/30
```

```
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2480 -
accuracy: 0.9058 - val_loss: 0.3013 - val_accuracy: 0.9180
Epoch 14/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2395 -
accuracy: 0.9078 - val_loss: 0.3550 - val_accuracy: 0.9050
Epoch 15/30
2240/2240 [==============================] - 6s 3ms/step - loss: 0.2303 -
accuracy: 0.9110 - val_loss: 0.3196 - val_accuracy: 0.9180
Epoch 16/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2225 -
accuracy: 0.9153 - val_loss: 0.3201 - val_accuracy: 0.9210
Epoch 17/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2166 -
accuracy: 0.9170 - val_loss: 0.3420 - val_accuracy: 0.9190
Epoch 18/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2093 -
accuracy: 0.9180 - val_loss: 0.3416 - val_accuracy: 0.9230
Epoch 19/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.2060 -
accuracy: 0.9202 - val_loss: 0.3826 - val_accuracy: 0.9190
Epoch 20/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1992 -
accuracy: 0.9234 - val_loss: 0.3563 - val_accuracy: 0.9240
Epoch 21/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1962 -
accuracy: 0.9238 - val_loss: 0.4002 - val_accuracy: 0.9200
Epoch 22/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1906 -
accuracy: 0.9249 - val_loss: 0.3840 - val_accuracy: 0.9270
Epoch 23/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1865 -
accuracy: 0.9275 - val_loss: 0.4120 - val_accuracy: 0.9140
Epoch 24/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1844 -
accuracy: 0.9273 - val_loss: 0.4259 - val_accuracy: 0.9170
Epoch 25/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1806 -
accuracy: 0.9295 - val_loss: 0.4514 - val_accuracy: 0.9280
Epoch 26/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1776 -
accuracy: 0.9306 - val_loss: 0.3949 - val_accuracy: 0.9240
Epoch 27/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1754 -
accuracy: 0.9308 - val_loss: 0.4218 - val_accuracy: 0.9170
Epoch 28/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1709 -
accuracy: 0.9322 - val_loss: 0.4251 - val_accuracy: 0.9220
Epoch 29/30
```

```
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1690 -
accuracy: 0.9327 - val_loss: 0.4300 - val_accuracy: 0.9260
Epoch 30/30
2240/2240 [==============================] - 7s 3ms/step - loss: 0.1654 -
accuracy: 0.9348 - val_loss: 0.4665 - val_accuracy: 0.9280
```

[44]:
```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 36)                37260

 dense_1 (Dense)             (None, 20)                740

 dense_2 (Dense)             (None, 4)                 84

=================================================================
Total params: 38,084
Trainable params: 38,084
Non-trainable params: 0
_________________________________________________________________
```

[45]:
```
#saving the trained model
model.save('saved_model')
```

```
INFO:tensorflow:Assets written to: saved_model\assets
```

[46]:
```
#reconstructing the saved model
reconstructed_model = keras.models.load_model('saved_model')
```

0.6 Generating Predictions

[47]:
```
#generating predictions
y_pred = np.argmax(reconstructed_model.predict(full_test_df), axis=1)

#converting encoded sentiments back to strings
y_pred_labels = ids_to_names(y_pred)
y_test_labels = ids_to_names(y_test)
```

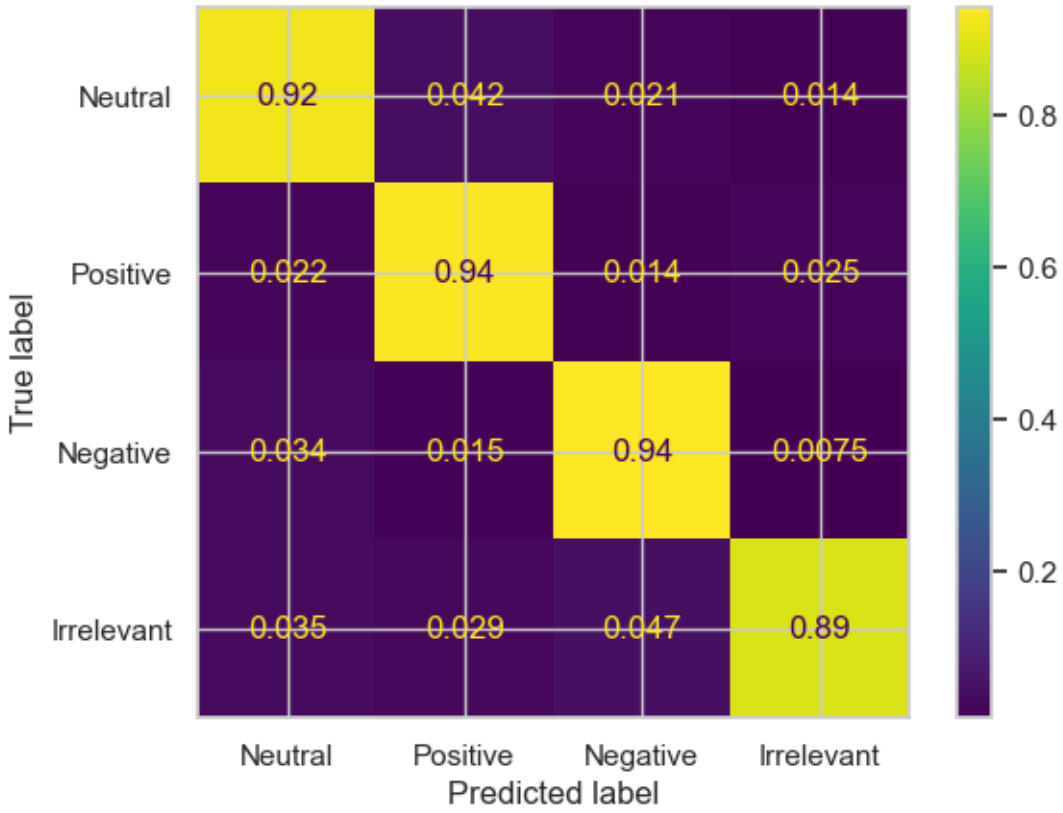
```
32/32 [==============================] - 0s 2ms/step
```

0.7 Visualising the Outputs

[48]:
```
#generating confusion matrix
y_unique = list(set(y_test_labels))
cm = confusion_matrix(y_test_labels, y_pred_labels, labels = y_unique,␣
 ↪normalize='true')
```

```
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=y_unique)
disp.plot()
```

[48]: <sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x1fd703d1040>



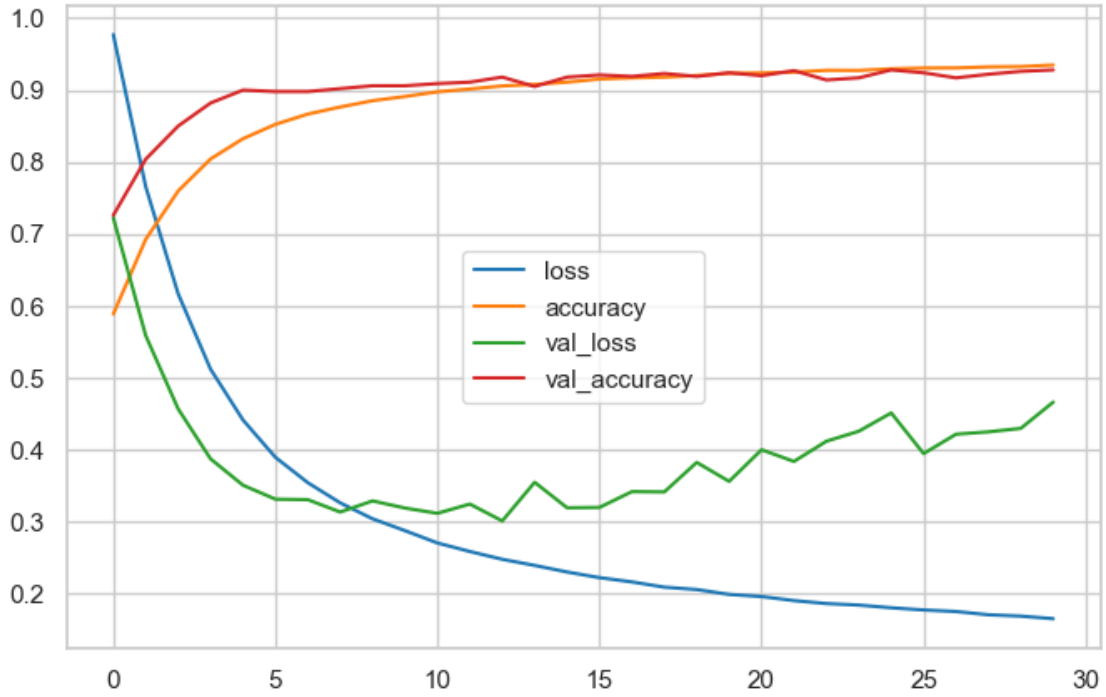
[49]:
```
#performance metrics
print(classification_report(y_pred_labels, y_test_labels))
```

```
              precision    recall  f1-score   support

  Irrelevant       0.89      0.92      0.91       166
    Negative       0.94      0.93      0.94       269
     Neutral       0.92      0.93      0.92       284
    Positive       0.94      0.93      0.93       281

    accuracy                           0.93      1000
   macro avg       0.92      0.93      0.92      1000
weighted avg       0.93      0.93      0.93      1000
```

[50]:
```
#accuracy plot
pd.DataFrame(h.history).plot(figsize=(8,5))
plt.show()
```



[51]:
```
#To see the final accuracy
accuracy_score(y_test, y_pred)
```

[51]: 0.927