from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

import matplotlib.pyplot as plt

import numpy as np

from tkinter import filedialog

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

from string import punctuation

from nltk.corpus import stopwords

import pandas as pd

from emoji import UNICODE\_EMOJI

main = tkinter.Tk()

main.title("NLP-based Extended Lexicon Model for Sarcasm Detection with Tweets and Emojis") #designing main screen

main.geometry("1300x1200")

sid = SentimentIntensityAnalyzer()

global filename

global dataset

global process

global sarcastic

global sentiment

def checkSarcasm(sentence):

pos = []

neg = []

neu = []

arr = sentence.split(' ')

for i in range(len(arr)):

word = arr[i].strip()

if word == 'smilingfacewithhearteyes':

word = 'excellent'

if word == 'loudlycryingface':

word = 'bad'

if word == 'winkingfacewithtongue':

word = 'happy'

if (sid.polarity\_scores(word)['compound']) >= 0.1:

pos.append(word)

elif (sid.polarity\_scores(word)['compound']) <= -0.1:

neg.append(word)

else:

neu.append(word)

return pos,neg,neu

def clean\_doc(doc):

tokens = doc.split()

table = str.maketrans('', '', punctuation)

tokens = [w.translate(table) for w in tokens]

tokens = [word for word in tokens if word.isalpha()]

stop\_words = set(stopwords.words('english'))

tokens = [w for w in tokens if not w in stop\_words]

tokens = [word for word in tokens if len(word) > 1]

tokens = ' '.join(tokens) #here upto for word based

return tokens

def upload():

global filename

global dataset

dataset = []

filename = filedialog.askopenfilename(initialdir="dataset")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n");

train = pd.read\_csv(filename,encoding='utf8',sep='\t')

count = 0

for i in range(len(train)):

tweet = train.get\_value(i,0,takeable = True)

print(tweet)

if str(tweet) != 'nan':

tweet = tweet.lower()

icon = train.get\_value(i,1,takeable = True)

if str(icon) != 'nan':

icon = UNICODE\_EMOJI[icon.strip()]

icon = ''.join(re.sub('[^A-Za-z\s]+', '', icon))

icon = icon.lower()

else:

icon = ''

msg = ''

if str(tweet) != 'nan':

arr = tweet.split(" ")

for k in range(len(arr)):

word = arr[k].strip()

if len(word) > 2:

msg+=word+" "

textdata = msg.strip()+" "+icon

#print(textdata)

dataset.append(textdata)

text.insert(END,'Total tweets found in dataset is : '+str(len(dataset)))

def Preprocessing():

text.delete('1.0', END)

global process

process = []

text.insert(END,'Messages after preprocessing and removing stopwords\n')

text.insert(END,'====================================================================================\n')

for i in range(len(dataset)):

sentence = dataset[i]

sentence = sentence.lower()

sentence = clean\_doc(sentence)

text.insert(END,sentence+'\n')

process.append(sentence)

A screenshot of a computer

Description automatically generated