import string  
from collections import Counter as cr  
import matplotlib.pyplot as plt  
  
# Read and clean the text  
text = open('read.txt', encoding='utf-8').read()  
lower\_case = text.lower()  
clean\_text = lower\_case.translate(str.maketrans('', '', string.punctuation))  
tokenized\_word = clean\_text.split()  
  
# Stop words list  
stop\_words = ["i", "me", "my", "myself", "we", "our", "ours", "ourselves", "you", "your", "yours", "yourself",  
 "yourselves", "he", "him", "his", "himself", "she", "her", "hers", "herself", "it", "its", "itself",  
 "they", "them", "their", "theirs", "themselves", "what", "which", "who", "whom", "this", "that", "these",  
 "those", "am", "is", "are", "was", "were", "be", "been", "being", "have", "has", "had", "having", "do",  
 "does", "did", "doing", "a", "an", "the", "and", "but", "if", "or", "because", "as", "until", "while",  
 "of", "at", "by", "for", "with", "about", "against", "between", "into", "through", "during", "before",  
 "after", "above", "below", "to", "from", "up", "down", "in", "out", "on", "off", "over", "under", "again",  
 "further", "then", "once", "here", "there", "when", "where", "why", "how", "all", "any", "both", "each",  
 "few", "more", "most", "other", "some", "such", "no", "nor", "not", "only", "own", "same", "so", "than",  
 "too", "very", "s", "t", "can", "will", "just", "don", "should", "now"]  
  
# Remove stop words  
final\_words = [word for word in tokenized\_word if word not in stop\_words]  
  
# Extract emotions  
emotion\_list = []  
with open('emotions.txt', 'r') as file:  
 for line in file:  
 clear\_line = line.replace("\n", '').replace(",", '').replace("'", '').strip()  
 if ':' in clear\_line:  
 word, emotion = clear\_line.split(':')  
  
 if word in final\_words:  
 emotion\_list.append(emotion)  
  
# Count emotions  
emotion\_count = cr(emotion\_list)  
total = sum(emotion\_count.values())  
  
# Calculate accuracy (percentage)  
emotion\_accuracy = {emotion: round((count / total) \* 100, 2) for emotion, count in emotion\_count.items()}  
  
# Plotting  
fig, ax = plt.subplots()  
emotions = list(emotion\_count.keys())  
counts = list(emotion\_count.values())  
  
# Add percentage to emotion labels  
emotion\_labels\_with\_percentage = [  
 f"{emotion} ({emotion\_accuracy[emotion]}%)" for emotion in emotions  
]  
  
bars = ax.bar(emotion\_labels\_with\_percentage, counts, color='skyblue')  
  
# Add count on top of each bar  
for i, bar in enumerate(bars):  
 height = bar.get\_height()  
 ax.text(bar.get\_x() + bar.get\_width()/2, height + 0.1,  
 f"{height}", ha='center', va='bottom', fontsize=9)  
  
# Graph formatting  
plt.title("Emotion Frequency with Accuracy (%)")  
plt.xlabel("Emotions (with %)")  
plt.ylabel("Counts")  
fig.autofmt\_xdate()  
plt.tight\_layout()  
plt.savefig('graph.png')  
plt.show()