MSc in Software Engineering

Τεχνικές Ανάλυσης Δεδομένων
Μέρας Γεώργιος
gmeras@uth.gr

Εργασία Εαρινού εξαμήνου 2025

Θέμα:

Ανάλυση Συναισθημάτων σε δεδομένα κειμένου

Περιεχόμενα

Σκοπός της εργασίας είναι η εξαγωγή και παρουσίαση των συναισθημάτων όπως αυτά αποτυπώνονται σε δεδομένα κειμένου. Στο πλαίσιο αυτής της εργασίας έχουμε αναπτύξει δύο μοντέλα και για τα τελικά αποτελέσματα χρησιμοποιήθηκε η Python. Τα μοντέλα που αναπτύχθηκαν στην εργασία ειναι το μοντόλο VADER και RoBERTa.

1) Το μοντέλο VADER (Valence Aware Dictionary and Sentiment Reasoner)

Για τη βασική ανάλυση συναισθήματος, χρησιμοποιήθηκε το VADER, ένα λεξικό που αξιολογεί τη συναθηματική φόρτηση βάσει λέξεων και γραμματικών κανόνων. Δεν αποτελέι μοντέλο μάθησης. Για το σκοπό αυτό χρησιμοποιήσαμε την βιβλιοθήκη nltk (Natural Language Toolkit).

Το VADER είναι ένα κομμάτι της παραπάνω βιβλιοθήκης και παρέχεται ως:

```
from nltk.sentiment import SentimentIntensityAnalyzer
```

2) Το μοντέλο RoBERTa

Το RoBERTa Model (Robustly Optimized BERT Approach) είναι ένα μοντέλο μηχανικής μάθησης για επεξεργασία φυσικής γλώσσας. Ανήκει στη βιβλιοθήκη Transformers και είναι εκπαιδευμένο. Υπάρχουν δίαφοροι τύποι του παραπάνω μοντέλου, σε αυτή την εργασία θα χρησιμοποιήσουμε το `"cardiffnlp/twitter-roberta-base-sentiment"` για την ανάλυση συναισθήματος σε τρία στάδια, αρνητικό, ουδέτερο, θετικό και το `'j-hartmann/emotion-english-distilroberta-base'` για Text-Classification, δηλαδή για κατηγοριοποίηση του συναισθήματος σε θυμό, χάρα, φόβο και άλλα. Περισσότερα μοντέλα υπάρχουν στο σύνδεσμο Models - Hugging Face .Σε κάθε μοντέλο πρέπει να του ανατίθεται το αντίστοιχο task. Θα χρησιμοποιήσουμε το πρώτο με task το `'sentiment-analysis'` και το δεύτερο με task `'text-classification'`.

Η εισαγωγή τους στην Python έγινε ως:

```
from  transformers import pipeline,AutoTokenizer, AutoModelForSequenceClassification
from scipy.special import softmax
```

1. Δεδομένα:

Χρησιμοποιήσαμε δεδομένα για τον εμβολιασμό του Κορωνοιού. Συγκεκριμένα τα δεδομένα αυτά έχουν 11.020 εγγραφές και σαν Attributes περιέχουν id, user_name, user_location, user_description, user_created, user_followers, user_friends, user_favourites ,user_verified, date, text, hastags, source, retweets, favorites, is_retweet.

Μετά την ανάλυση του συναισθήματος προσθέσαμε στο ηδη υπάρχον DataFrame τα εξής Attributes:

- emotion: Λίστα με όλα τα συναισθήματα
- main_emotion: Το μέγιστο από το emotion
- roberta_neg: Πιθανότητα αρνητικού συναισθήματος με μηχανική μάθηση

- roberta_neu: Πιθανότητα ουδετέρου συναισθήματος με μηχανική μάθηση
- roberta_pos: Πιθανότητα θετικού συναισθήματος με μηχανική μάθηση
- predicted_sentiment: Θετικό, αρνητικό ή ουδέτερο ανάλογα με τις παραπάνω πιθανοτητες
- neg: Αρνητικό με μοντέλο Vader
- neu: Ουδέτερο με μοντέλο Vader
- pos: Θετικό με μοντέλο Vader
- compouund: Το συνολικό score από -1 εως 1, όσο κοντίτερα στο -1 τόσο πιο αρνητικό συναίσθημα.
- sentiment: Αρνητικό, Θετικό, Ουδέτερο συναίσθημα με βάση το παραπάνω score.

2. Καθαρισμός Δεδομένων:

Ο καθαρισμός δεδομένων έγινε με μια συνάρτηση cleanText(text) με την οποία έπρεπε να αφαιρέσουμε συνδέσμους και αναφορές. Δεν μας ενδιέφερε επίσης οτιδήποτε δεν ήταν αλφαριθμητικό. Για να αφαιρέσουμε τα παραπάνω χρησιμοποιήσαμε τη βιβλιοθήκη Regular Expression - re.

```
def cleanText(text):
    text = re.sub(r'http\S+',' ',text)
    text = re.sub(r'www\.\S+',' ',text)
    text = re.sub(r'#\w+',' ',text)
    text = re.sub(r'@\w+', ' ', text)
    text = re.sub(r'[^a-zA-Z0-9]', ' ', text)
    return text
```

3. Εξαγωγή συναισθημάτων:

Για την εξαγωγή συναισθημάτων χρησιμοποιήθηκαν οι παρακάτω βιβλιοθήκες:

```
import  pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.sentiment import SentimentIntensityAnalyzer
from tqdm import tqdm
import re
from transformers import pipeline,AutoTokenizer, AutoModelForSequenceClassification
from scipy.special import softmax
```

Η pandas ήταν η βιβλιοθήκη η οποία μας βοήθησε να διαβάσουμε από μορφή CSV το αρχείο με τα δεδομένα, επίσης φτιάξαμε νέα DataFrames τα οποία ενσωματώσαμε στο τελικό αρχείο.

Οι βιβλιοθήκες matplotlib και seaborn ήταν εκείνες με τις οποίες κάναμε τα τελικά γραφήματα.

Η tqdm μας επέτρεψε να κάνουμε loop μέσα στο DataFrame της pandas.

Οι nltk και tranformers είναι οι βιβλιοθήκες εξαγωγής συναισθήματος, με την πρώτη να έχει ένα συνολικό score ανάλογα με τη ρίζα της κάθε λέξης, ενώ από τη δεύτερη έχουμε ένα μοντέλο μηχανικής μάθησης, που αξιολογεί κάθε πρόταση και επιστρέφει ένα διάνυσμα πιθανότητας.

Από τη scipy υπολογίσαμε την πιθανότητα του Roberta model.

1. Εξαγωγή Συναισθήματος με VADER

Στο παρακάτω for μπαίνουμε στο DataFrame και εφαρμόζουμε την cleanText(text) για κάθε text, ώστε να αφαιρεθούν οι περιττές πληροφορίες που υπήρχαν, όπως αναφορές σε εταιρείες, hastags και ιστοσελίδες. Το κείμενο καθαρίστηκε, καθώς εκφράσεις όπως οι προηγούμενες, κάνουν το κείμενο ουδέτερο.

```
sia = SentimentIntensityAnalyzer()
res = {}
res_rob = {}
roberta_res = {}
for i,row in tqdm(df.iterrows(), total=len(df)):
    text = cleanText(row['text'])
    myid = row['id']
    res[myid] = sia.polarity_scores(text)
    res_rob[myid] = emotion_model(text)
    roberta_res[myid] = Roberta_polarity_score(text)
```

Η μέθοδος polarity_scores() καλείται πάνω στο αντικείμενο sia, το οποίο είναι αντικείμενο της κλάσης SentimentIntensityAnalyzer() και επιστρέφει ένα λεξικό με θετικό, αρνητικό, ουδέτερο συναίσθημα και ένα τελικό score αυτών (compound), με τιμές απο -1 εως 1. Για τιμές από [-1 , -0.05) θεωρούμε το κείμενο, κείμενο με αρνητικό συναίσθημα. Αν το compound βρίσκεται στις τιμές (0.05 , 1], θεωρούμε το κείμενο, κείμενο με θετικό συναίσθημα και ουδέτερο αλλίως. Το αποθηκέυουμε σε ένα λεξικό επίσης με key=myid ώστε να γνωρίζουμε το συναίσθημα κάθε χρήστη βάσει του id του. Τελικά φτίαξαμε μια ακόμη στήλη με βάση τα παραπάνω, η οποία “βάφτιζε” κάθε χρήστη σε θετικό αρνητικό ή ουδέτερο. Την συνάρτηση αυτή την ονομάσαμε pos() και φαίνεται παρακάτω.

```
def pos(score):
    if score >= 0.05:
        return 'Positive'
    elif score < 0.05 and score >= -0.05:
        return 'Neutral'
    else:
        return 'Negative'
```

Την παραπάνω συνάρτηση την εφαρμόσαμε πάνω στη στήλη compound και φτίαξαμε μια στήλη “sentiment”.

```
pf = pd.DataFrame(res).T
pf = pf.reset_index().rename(columns={'index':'id'})
pf['sentiment'] = pf['compound'].apply(pos)
pf = pf.merge(df, how='left', on='id')
```

neg	neu	pos	compound	sentiment
0	0.748	0.252	0.4019	Positive
0.125	0.766	0.109	-0.1027	Negative
0	0.75	0.25	0.25	Positive
0	1	0	0	Neutral

Απόσπασμα του VADER model.

Κάνουμε μια εκπτύπωση του θετικού συναιθήματος “Positive” για min και max compound, δηλαδή για τα δύο άκρα.

```
pd.set_option('display.max_colwidth', None)
df = df[df['sentiment']=='Positive']
print('Max from positive: '+df[df['compound'] == df['compound'].max()]['text'])
print('Min from positive: '+df[df['compound'] == df['compound'].min()]['text'])
```

Το αποτέλεσμα του παραπάνω ήταν το εξής:

```
9037    Max from positive: Thrilled, SO relieved, and VERY thankful to get second shot of #PfizerBioNTech #Covid19vaccine in-home today -- YAY…
Name: text, dtype: object
1801    Min from positive: Really excited to get my #covidvacccine today. Had to delay mine for a week cuz I was on service, but couldnt be pr…
Name: text, dtype: object
```

Αποτέλεσμα VADER για μέγιστο και ελάχιστο θετικό συναίσθημα

Να σημειωθεί πως παραπάνω φαίνεται το κείμενο στην αρχική του μορφή, (όχι καθαρισμένο) καθώς η cleanText() επιστρέφει ένα text μόνο για επεξεργασία και όχι για αποθήκευση στο αρχικό μας κείμενο.

2. Εξαγωγή συναισθήματος με RoBERTa

Το μοντέλο RoBERTa μας βοήθησε να κάνουμε ανάλυση σε δύο επιμέρους επίπεδα. Το πρώτο είναι η ανάλυση συναισθήματος σε αρνητικό, θετικό και ουδέτερο, το οποίο έγινε με `"cardiffnlp/twitter-roberta-base-sentiment"`, και το δέυτερο είναι η ποικιλία των συναισθημάτων, το οποίο έγινε με `'j-hartmann/emotion-english-distilroberta-base'`. Η επιλογή των μοντέλων έγινε βάσει του Validation του κάθε μοντέλου.

Η ανάθεση των μοντέλων έγινε ως:

```
#Roberta pretrain model for sentiment analysis {fear,joy,anger..}
emotion_model = pipeline('text-classification',model='j-hartmann/emotion-english-distilroberta-base',top_k=None)

#Roberta pretrain model for sentiment analysis pos,neu,neg
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
```

```
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
sent_pipeline = pipeline('sentiment-analysis',model=model,tokenizer=tokenizer)
```

Θα χρειαστούμε μια συνάρτηση για να εξάγουμε την πιθανότητα θετικού, αρνητικού και ουδέτερου συναισθήματος.

```
def Roberta_polarity_score(text):
    encoded_text = tokenizer(text, return_tensors='pt')
    output = model(**encoded_text)
    scores = output[0][0].detach().numpy()
    scores = softmax(scores)
    res_pretrain = {'roberta_neg': scores[0], 'roberta_neu': scores[1], 'roberta_pos': scores[2]}
    return res_pretrain
```

Η παραπάνω συνάρτηση μετατρέπει το text σε ένα διάνυσμα, ανάλογα με το πώς θεωρεί το μοντέλο (tokenizer) σωστό να χωριστεί. Παραθέτουμε παράδειγμα παρακάτω.

```
example = 'I love pizza'
Roberta_polarity_score(example)
```

Το αποτέλεσμα ήταν:

```
input_ids': tensor([[   0,  100,  657, 9366,    2]]), 'attention_mask': tensor([[1, 1, 1, 1, 1]])}
quenceClassifierOutput(loss=None, logits=tensor([[-1.9922, -0.6216,  3.4267]], grad_fn=<AddmmBackward0>), hidden_states=None, attentions=None
1.9921911 -0.621593   3.4267075]
```

Παράδειγμα RoBERTa.

Τα input_ids είναι τα μοναδικά αναγνωριστηκά για κάθε λέξη, πχ I=100, Love=657, Pizza=9366, 0 και 2 είναι αρχή και τέλος. Το attention_mask είναι τα σημεία που πρέπει το μοντέλο να δώσει έμφαση. Τέλος βλέπουμε τρία logits τα οποία αντιστοιχούν σε αρνητικό, ουδέτερο και θετικό συναίσθημα. Πάνω στα logits εφαρμόζουμε τη sofmax() για να πάρουμε την πιθανότητα, αφού πούμε στην python ποιο μέρος του output να αποκόψει (output[0][0].detach()). Αφού έχουμε πάρει την πιθανότητα για κάθε id μέσω του for που είδαμε πρίν εδώ. Φτίαχνουμε ξανά ένα DataFrame το οποίο και ενσωματώνουμε στο προηγούμενο, μαζί με μια στήλη predicted_sentiment η οποία περιέχει το συναίσθημα σαν Positive,Negative και Neutral. Για τη στήλη αυτή, κρατήσαμε το id με την αντίστοιχη μέγιστη τιμή της πιθανότητας, όπως φαίνεται παρακάτω.

```
label_map = {
  'roberta_neg':'NEGATIVE',
  'roberta_neu':'NEUTRAL',
  'roberta_pos':'POSITIVE',
}
pretrain_df = pd.DataFrame(roberta_res).T
pretrain_df = pretrain_df.reset_index().rename(columns={'index':'id'})
pretrain_df['predicted_sentiment'] = pretrain_df[['roberta_neg','roberta_neu','roberta_pos']].idxmax(axis=1)
pretrain_df['predicted_sentiment'] = pretrain_df['predicted_sentiment'].map(label_map)
pretrain_df = pretrain_df.merge(pf, how='left', on='id')
robert_df = robert_df.merge(pretrain_df, how='left', on='id')
```

3. Εξαγωγή ποικίλων συναισθημάτων με RoBERTa

Για να εξάγουμε τα ποικίλα συναισθήματα με το RoBERTa χρησιμοποιήσαμε:

```
#Roberta pretrain model for sentiment analysis {fear,joy,anger..}
emotion_model = pipeline('text-classification',model='j-hartmann/emotion-english-distilroberta-base',top_k=None)
```

Έχουμε αναθέσει το task: text classification και top_k=None. Το πρώτο είναι η εργασία που θέλω να κάνει και το δεύτερο τα κορυφαία συναισθήματα που θα βγάλει. Με None δεν έχουμε κανέναν περιορισμό. Οπως φαίνεται στο for εδώ, αποθηκέυουμε ένα λεξικό μέσα σε ένα άλλο, το οποίο περιέχει όλα τα συναισθήματα. Φτιάχνουμε ένα DataFrame και το ενσωματώνουμε στο προηγούμενο για την τελική ανάλυση. Δημιουργούμε επίσης μια στήλη με όνομα main_emotion η οποία έχει το μέγιστο συναίσθημα.

```
robert_df = pd.DataFrame(res_rob).T
robert_df = robert_df.reset_index().rename(columns={'index':'id',0:'emotion'})
robert_df['main_emotion'] = robert_df['emotion'].apply(lambda x: max(x,key=lambda item: item['score'])['label'])
```

4. Κατηγοριοποίηση των χρηστών με βάση το συναίσθημα που εκφράζουν:

1. Σε ηπείρους

Κατεβάσαμε επίσης δεδομένα με πόλεις, χώρες και περιοχές, με τα οποία κατηγοριοποιήσαμε αρχικά τους χρήστες σε Ηπείρους βάσει του user_location.

```
lookup = {}
for _, row in regions.iterrows():
    try:
        city = str(row['city_ascii']).lower()
        country = str(row['country_name']).lower()
        region_value = row['region']
        lookup[city] = region_value
        lookup[country] = region_value
    except:
        continue

def region(location):
    location = str(location).lower()
    for keyword in lookup:
        if keyword in location:
            return lookup[keyword]
    return 'Other'

df1 = df[df['main_emotion'] == 'fear']
df1 = df1[~df1['user_location'].isnull()]
df1['region'] = df1['user_location'].apply(region)
```

Χρησιμοποιούμε ένα try block, καθώς στο αρχείο region μπορεί να προκύψουν, πόλεις χωρίς χώρα ή πόλεις χωρίς ήπειρο κλπ. Η κατηγοριοποίηση των χρηστών, γίνεται με τη συνάρτηση region(), εφαρμόζοντάς τη πάνω στη στήλη user_location.

Η παραπάνω συνάρτηση ελέγχει αν υπάρχει χώρα ή πόλη, που να αντιστοιχεί σε κάποια ήπειρο. Στη στήλη user_location υπήρχαν δεδομένα, τα οποία δεν είχαν σχέση με γεωδαισία, όπως για παράδειγμα “not today” και άλλα. Στην συνάρτηση region(), δεν χρειάζεται να γίνει κάποιο strip, καθώς αν το keyword υπάρχει στο location, με οποιαδήποτε μορφή (πχ USAus) η Python θα επιστρέψει True.

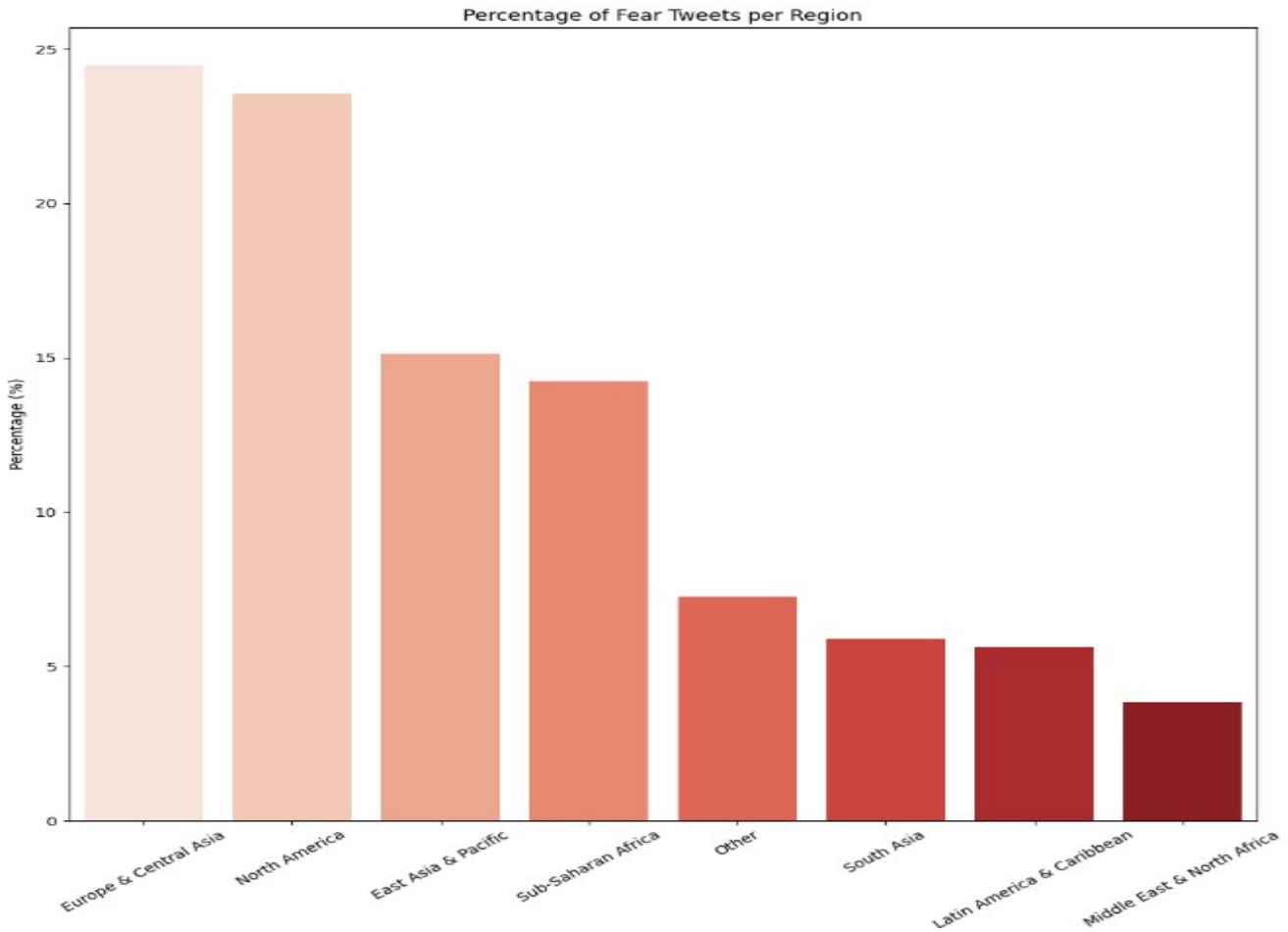

Figure 1: Ηπειροι και ποσοστό έκφρασης φόβου.

2. Με βάση τους Followers

Κατηγοριοποιήσαμε τους χρήστες σε “famous” και “infamous” ανάλογα με τους followers και την επικύρωση του λογαριασμού τους (user_verified = True). Θεωρήσαμε σαν famous χρήστες αυτούς με πάνω από 10.000 followers και infamous αλλιώς. Ο σκοπός είναι να πάρουμε μια εικόνα, για την μεροληψία, υπέρ ή κατά του εβμολίου. Χρησιμοποιήσαμε την παρακάτω συνάρτηση.

```
def famous(followers):
    if followers>10000:
        return 'famous'
    else:
        return 'infamous'
df2 = df[df['user_verified']==True]
df2['famous'] = df2['user_followers'].apply(famous)
```

Οι χρήστες φαίνεται να κατανέμονται ισόποσα, με τον φόβο να εκφράζεται σε ελάχιστα μεγαλύτερα επίπεδα.

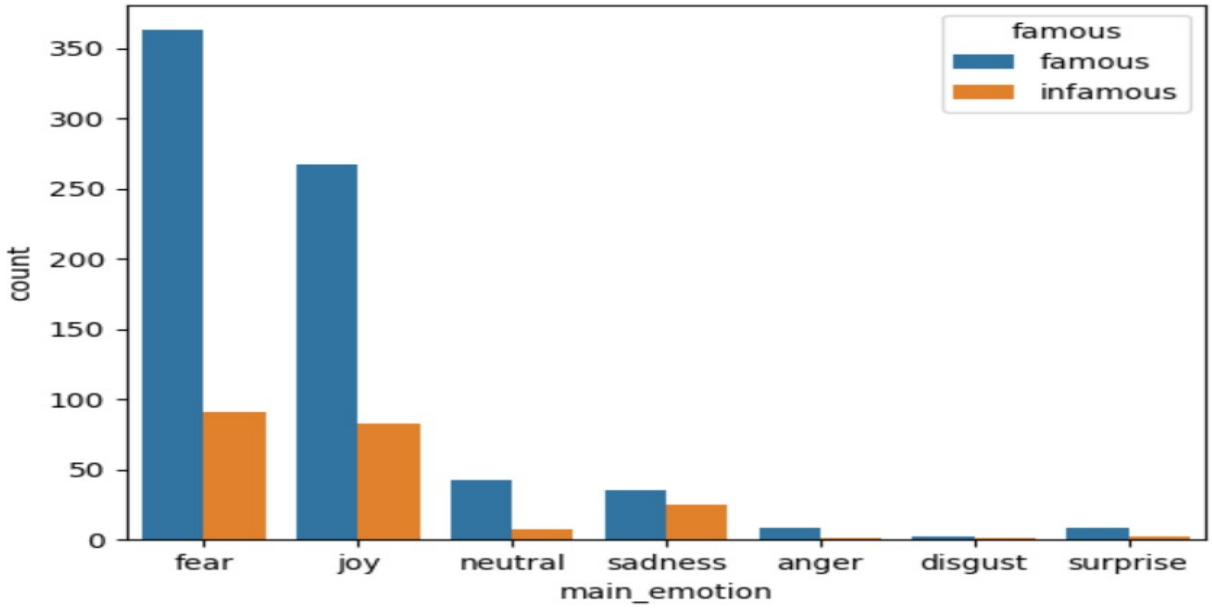

Figure 2: Μεροληψία υπερ ενός

3. Με βάση την επικύρωση του λογαριασμού τους.

Κατηγοριοποιήσαμε επίσης τους χρήστες σε Verified_False και Verified_True για να δούμε το συναίσθημα του κάθενος. Η κατηγοριοποίηση εδώ ήταν πιο έυκολη καθώς είχαμε εξ' αρχής καλά δεδομένα.

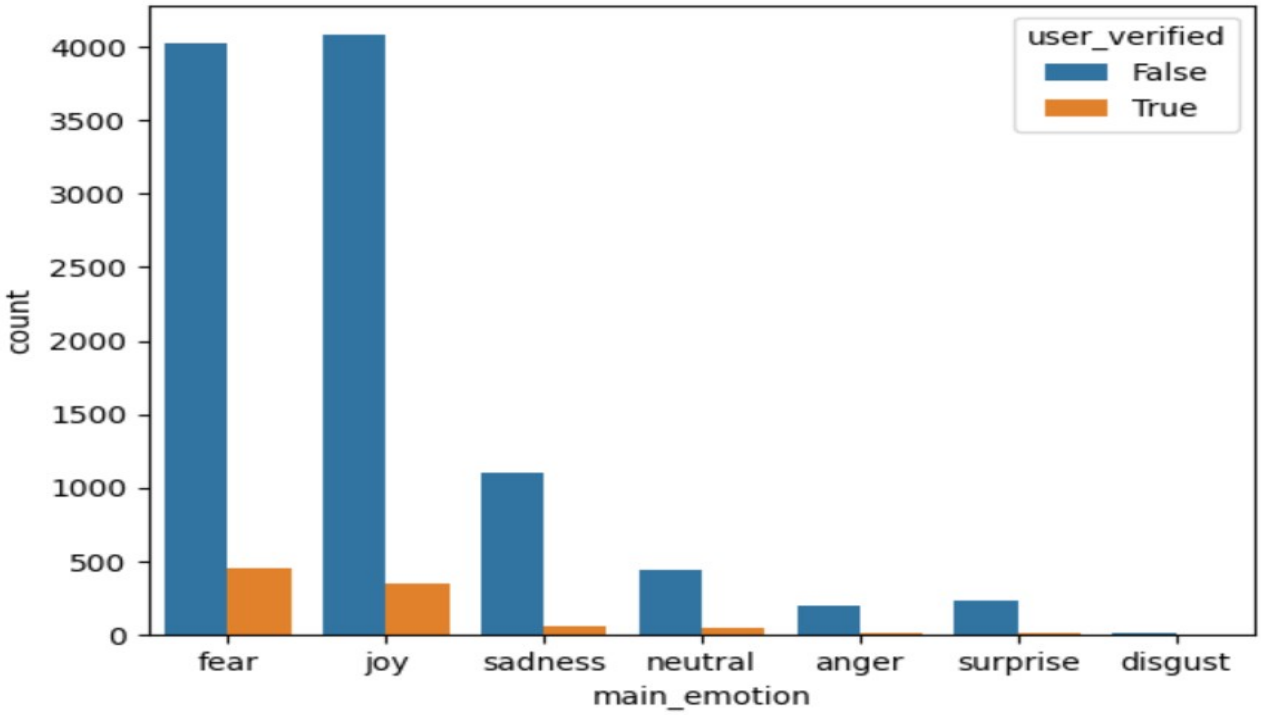

Figure 3: Συναίσθημα που εκφράζουν οι χρήστες ανάλογα με την ταυτότητά τους

5. Συνολικό συναίσθημα

1.Συναίσθημα βάσει VADER

Παρατηρούμε, βάσει του μοντέλου VADER, πως το θετικό συναίσθημα, είναι μεγαλύτερο από το αρνητικό.

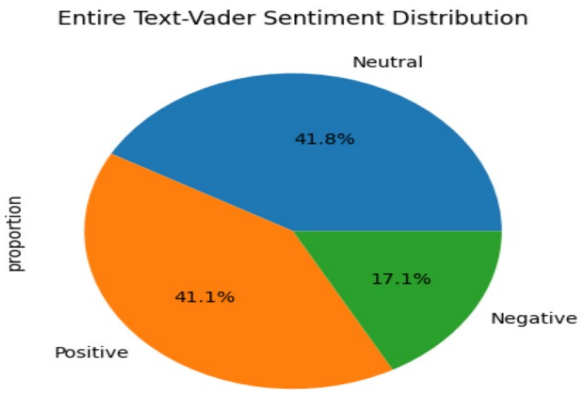

Figure 4: Συνολικό συναίσθημα VADER

Υπάρχει συσχέτιση μεταξύ VADER και RoBERTa model;

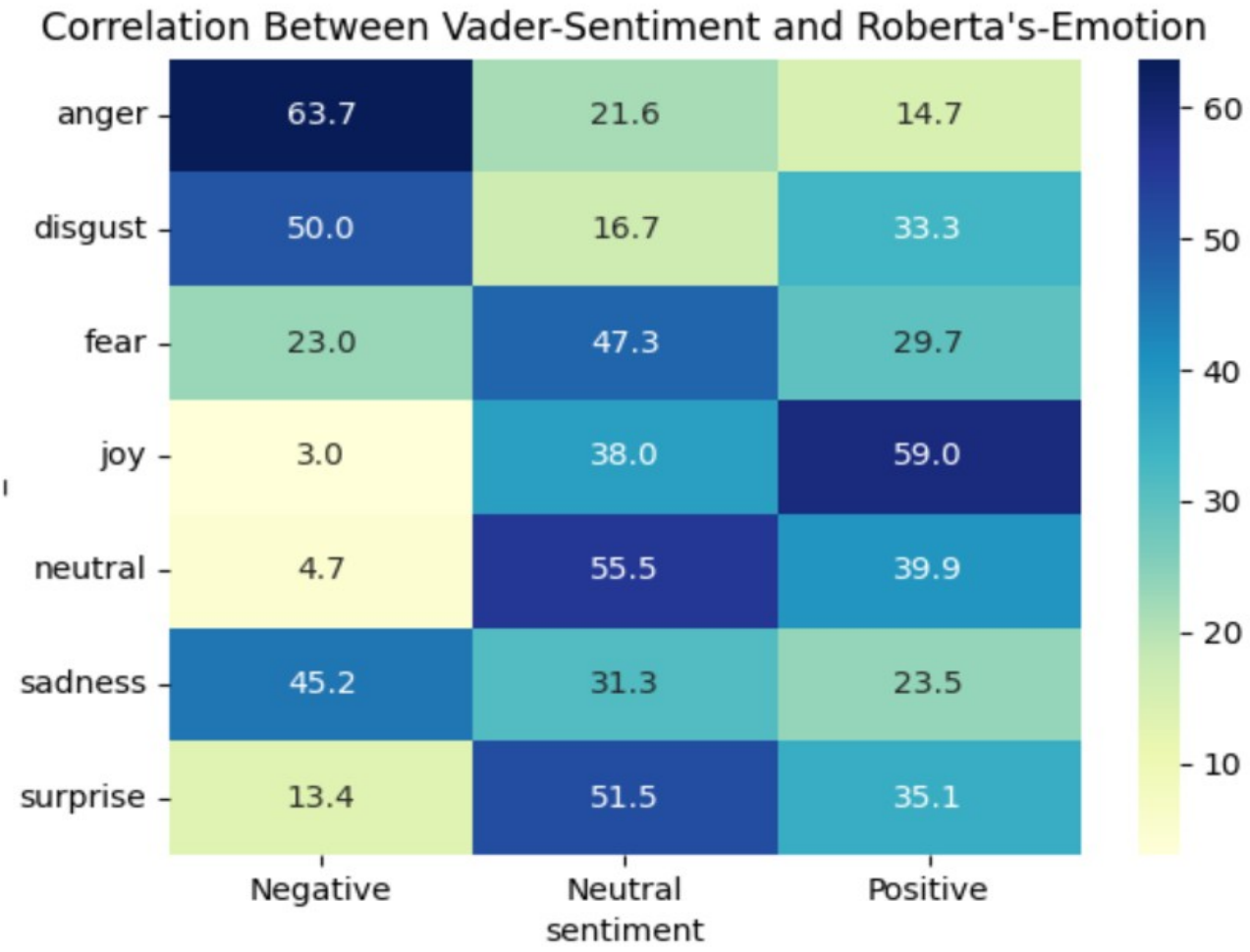

Figure 5: Συσχέτιση μεταξύ VADER και RoBERTa

Παρατηρούμε βάσει του παραπάνω πως τα αρνητικά συναισθήματα ανήκουν σε μεγαλύτερο κομμάτι στο Negative του VADER και τα θετικά στο Positive, εκτός από το συναίσθημα του φόβου. Αυτό σημαίνει πως το VADER, περιεχόμενο με φόβο, το έχει κατατάξει σε θετικό συναίσθημα ή το αντίστροφο.

2.Συναίσθημα βάσει RoBERTa

Με το μοντέλο RoBERTa, εξάγαμε τα ποικίλα συναισθήματα, αλλά και το θετικό, αρνητικό ή ουδέτερο συναίσθημα που εκφράζουν οι χρήστες. Παρακάτω φαίνονται τα δύο διαγράμματα.

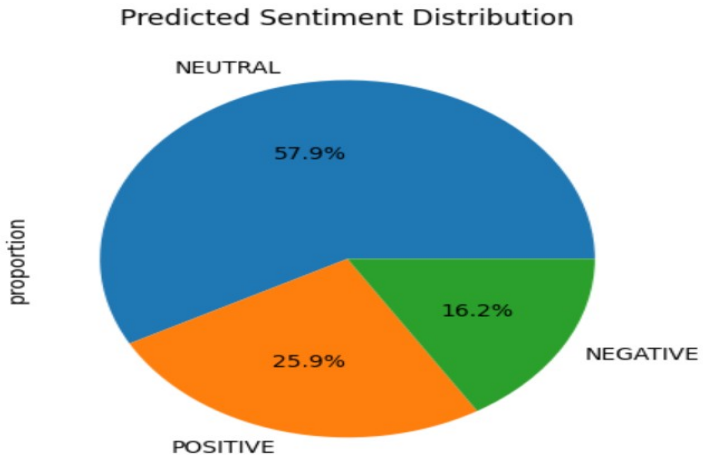

Figure 6: Συναίσθημα με RoBERTa

Φαίνεται πως το RoBERTa επίσης εξάγει ένα θετικό συναίσθημα στους χρήστες, όχι όμως στο ίδιο ποσοστό του VADER.

Τα ποσοστά των ποικίλων συναισθημάτων απεικονίζονται παρακάτω.

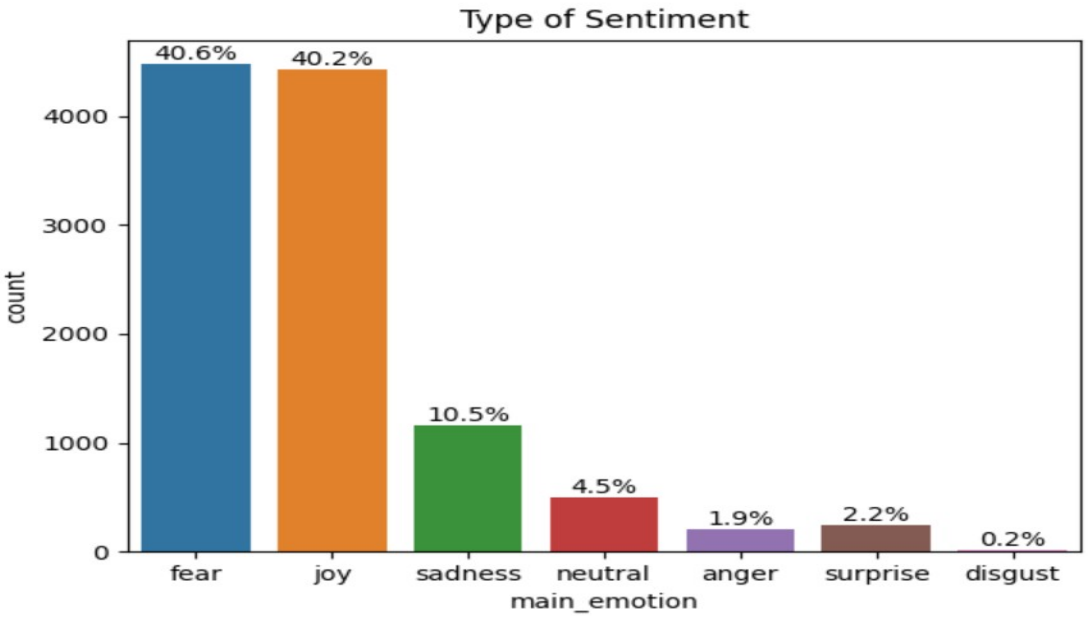

Figure 7: Ποσοστά συναισθημάτων

6. Συμπεράσματα

- Το κείμενο είτε πρόκειται για ανάλυση συναισθήματος με VADER είτε με RoBERTa το αποτέλεσμα είναι υπέρ του θετικού.
- Τα ποικίλα συναισθήματα είναι ο φόβος, η χαρά, η λύπη, ο θυμός, η έκπληξη και η απέχθεια με το μεγαλύτερο ποσοστό στο φόβο.
- Οι ηπειροι που εκφράζουν τον μεγαλύτερο φόβο είναι οι Ευρώπη και κεντρική Ασια και η βόρεια Αμερική.
- Οι χρήστες που ονομάσαμε “famous” δεν φαίνεται να έχουν μεροληψία υπερ ή κατά του εμβολίου.

Σημείωση: Τα δεδομένα χρησιμοποιήθηκαν για μια βασική ανάλυση συναισθήματος και κατηγοριοποίηση. Δεν θέλουμε και δεν μπορούμε, να εξάγουμε ευαίσθητα συμπεράσματα, καθώς ο αριθμός των δεδομένων είναι πολύ μικρός. Η εργασία δεν έχει σκοπό να προσβάλει καμία ομόδα ανθρώπων.

Μέρας Γεώργιος
MSc Candidate in Software Engineering
Πανεπιστήμιο Θεσσαλίας
Τμήμα Ψηφιακών Συστημάτων
gmeras@uth.gr