ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΔΙΟΙΚΗΣΗ ΚΑΙ ΨΗΦΙΑΚΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ»

“ΜΕΛΕΤΗ TWITTER API ΚΑΙ ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ”

Διπλωματική εργασία

Ιωσήφ Κωνσταντουράκης, ΜΔΕ01

Επιβλέποντας: Μαστοράκης Γεώργιος, Αναπληρωτής Καθηγητής

Άγιος Νικόλαος, Απρίλιος 2023

ΕΛΛΗΝΙΚΟ ΜΕΣΟΓΕΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΣΧΟΛΗ ΕΠΙΣΤΗΜΩΝ ΔΙΟΙΚΗΣΗΣ ΚΑΙ ΟΙΚΟΝΟΜΙΑΣ
ΤΜΗΜΑ ΔΙΟΙΚΗΤΙΚΗΣ ΕΠΙΣΤΗΜΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

«ΔΙΟΙΚΗΣΗ ΚΑΙ ΨΗΦΙΑΚΟΣ ΜΕΤΑΣΧΗΜΑΤΙΣΜΟΣ»

“ΜΕΛΕΤΗ TWITTER API ΚΑΙ ΑΝΑΛΥΣΗ ΔΕΔΟΜΕΝΩΝ”

Τριμελής Εξεταστική Επιτροπή

Ονοματεπώνυμο:	Μαστοράκης Γεώργιος
Βαθμίδα:	Αναπληρωτής Καθηγητής
Τμήμα, Ίδρυμα:	Διοικητικής Επιστήμης και Τεχνολογίας, Ελληνικό Μεσογειακό Πανεπιστήμιο

Ονοματεπώνυμο:	Κοπανάκης Ιωάννης
Βαθμίδα:	Καθηγητής
Τμήμα, Ίδρυμα:	Διοικητικής Επιστήμης και Τεχνολογίας, Ελληνικό Μεσογειακό Πανεπιστήμιο

Ονοματεπώνυμο:	Περακάκης Εμμανουήλ
Βαθμίδα:	Επίκουρος Καθηγητής
Τμήμα, Ίδρυμα:	Διοικητικής Επιστήμης και Τεχνολογίας, Ελληνικό Μεσογειακό Πανεπιστήμιο

HELLENIC MEDITERRANEAN UNIVERSITY
SCHOOL OF MANAGEMENT AND ECONOMICS SCIENCE
DEPARTMENT OF MANAGEMENT SCIENCE AND TECHNOLOGY

MSc in

MANAGEMENT AND DIGITAL TRANSFORMATION

“STUDY OF TWITTER API AND DATA ANALYSIS”

MASTER THESIS

Iosif Konstantourakis, MDE01

Supervisor: Mastorakis George, Associate Professor

Agios Nikolaos, April 2023

Υπεύθυνη Δήλωση : Βεβαιώνω ότι είμαι συγγραφέας αυτής της διπλωματικής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στη διπλωματική εργασία. Επίσης έχω αναφέρει τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε παραφρασμένες. Επίσης βεβαιώνω ότι αυτή η διπλωματική εργασία προετοιμάστηκε από εμένα προσωπικά ειδικά για τις απαιτήσεις του Μεταπτυχιακού Προγράμματος Σπουδών “Διοίκηση και Ψηφιακός Μετασχηματισμός” του Τμήματος Διοικητικής Επιστήμης και Τεχνολογίας του ΕΛΜΕΠΑ.

Ευχαριστίες

Πρώτιστα και κύρια, νιώθω την ανάγκη να ευχαριστήσω την οικογένειά μου, για την αμέριστη στήριξή και την παρότρυνσή τους σε όλη την διάρκεια αυτού του μεταπτυχιακού.

Κατόπιν, τον φίλο και συνάδελφο Γιάννη Καπανταϊδάκη για τις κατευθύνσεις και τις συμβουλές του κατά την εκπόνηση της διπλωματικής.

Και τέλος, τον επιβλέποντα καθηγητή μου Γιώργο Μαστοράκη, για τις αρχικές κατευθύνσεις που μου έδωσε, τις παρατηρήσεις του και την γενικότερη συνεισφορά του στον τρόπο που έπρεπε να κινηθώ για την συγγραφή αυτής της διπλωματικής εργασίας.

Περίληψη

Στα πλαίσια της επιστήμης του Marketing, η κατανόηση των τάσεων και των απόψεων γύρω από θεματικές, γεγονότα, πρόσωπα και προϊόντα είναι μεγάλης σημασίας, και η συναισθηματική ανάλυση κειμένων αποτελεί ένα πολύ χρήσιμο εργαλείο για αυτόν τον σκοπό. Η επικρατέστερη μέθοδος τα τελευταία χρόνια ήταν η χρησιμοποίηση λεξικών συναισθηματικά προ-αξιολογημένων όρων. Όταν τα λεξικά αυτά είναι διαθέσιμα και επαρκώς αναλυτικά (όπως για παράδειγμα στην αγγλική γλώσσα), τα αποτελέσματά τους μπορούν να είναι εξαιρετικά. Όταν όμως δεν είναι όσο εκτεταμένα/εξειδικευμένα χρειάζεται ή σε κάποιες γλώσσες (όπως η ελληνική) μπορεί να είναι πολύ περιορισμένα έως και ανύπαρκτα, είναι αναγκαία μια εναλλακτική προσέγγιση που θα μπορεί να αποδίδει ικανοποιητικά αποτελέσματα συναισθηματικής ανάλυσης έχοντας διαθέσιμα μόνο πρωτογενή, μη προ-αξιολογημένα κείμενα.

Η παρούσα διπλωματική εργασία, αναφορικά με το κομμάτι της ανάλυσης δεδομένων, εξετάζει την εφικτότητα και κάνει μια πρώτη διερεύνηση της αποτελεσματικότητας της χρήσης σύγχρονων μοντέλων μηχανικής μάθησης (machine learning), προεκπαιδευμένων σε εργασίες συμπερασμάτων φυσικής γλώσσας (Natural Language Inference - NLI), για την ανάλυση συναισθήματος συγκεκριμένα σε ελληνικά κείμενα, μέσω τεχνικών ταξινόμησης μηδενικής βολής (zero-shot classification). Για τον σκοπό αυτό, το αρχικό τμήμα της εργασίας ασχολείται με το κοινωνικό δίκτυο Twitter και τις προγραμματιστικές διεπαφές του για την συλλογή των κειμένων προς αξιολόγηση, και κατόπιν με τεχνικές προεπεξεργασίας τους.

Μια πρώτη απόπειρα βελτίωσης των δυνατοτήτων κατανόησης των χρησιμοποιούμενων μοντέλων, μέσω εμπλουτισμού των λεξικών τους με λέξεις από τα συλλεγμένα κείμενα, αποδεικνύεται αναποτελεσματική όταν δεν ακολουθείται από επανεκπαίδευσή τους - διαδικασία “ακριβή” από υπολογιστικής άποψης, που ξεφεύγει από τους σκοπούς της εργασίας αυτής. Χρησιμοποιώντας λοιπόν χωρίς μεταβολές ελεύθερα διαθέσιμα προεκπαιδευμένα πολυγλωσσικά μοντέλα NLI (στην περίπτωσή μας το DeBERTa) πάνω σε ελληνικά tweets, διαπιστώνεται ότι τα αποτελέσματά τους για σκοπούς ανάλυσης συναισθήματος σε πρωτογενή κείμενα, είναι πολύ ενθαρρυντικά. Η διπλωματική εργασία κλείνει προτείνοντας μια επιπλέον μέθοδο αξιολόγησης των αποτελεσμάτων, καθώς και πεδία πιθανής βελτίωσης της μεθόδου προς τους επόμενους ερευνητές.

Λέξεις-κλειδιά: Διεπαφή προγραμματισμού εφαρμογών, Συναισθηματική ανάλυση, Συμπεράσματα φυσικής γλώσσας, Μετασχηματιστές, Αναγνώριση γλωσσικών μονάδων, Ταξινόμηση μηδενικής βολής

Abstract

In the context of Marketing science, understanding trends and opinions regarding topics, events, people and products is of great importance, and sentiment analysis of texts is a very useful tool towards this goal. The predominant method of such analysis in recent years has been the use of dictionaries of emotionally pre-evaluated terms. Whenever these dictionaries are available and sufficiently detailed (as for example in the English language), their results can be excellent. However, when they are not as extensive/specialized as needed or in some languages (such as Greek) are very limited or even non-existent, an alternative approach is necessary that will be able to yield satisfactory emotional analysis results, making only use of raw, non-evaluated texts.

This thesis, regarding the part of data analysis, examines the feasibility and makes an initial investigation of the effectiveness of the use of modern machine learning models, pre-trained in Natural Language Inference (NLI) tasks, for sentiment analysis specifically in Greek texts, through zero-shot classification techniques. For this purpose, the initial part of the work deals with the Twitter social network and its programming interfaces (APIs) for the collection of the texts to be evaluated, and then with their pre-processing techniques.

A first attempt to improve the understanding capabilities of the models used, by expanding their dictionaries with words from collected texts, proves to be ineffective when not followed by model retraining - a computationally expensive process, which is beyond the scope of this work. Using unchanged freely available pre-trained multilingual NLI models (in our case DeBERTa) on Greek tweets, it is found that their results regarding sentiment analysis purposes on raw texts, are very encouraging. The thesis concludes by proposing an additional method of results evaluation, as well as fields of possible improvement of the method for the future researchers.

Keywords: Application programming interface, Sentiment analysis, Natural Language Inference, Transformers, Tokenization, Zero-shot classification

Περιεχόμενα

Λίστα εικόνων

Λίστα πινάκων

1. Εισαγωγή

Στα γενικότερα πλαίσια του marketing, χρειάζεται να γνωρίζουμε ανά πάσα στιγμή τον αντίκτυπο που έχει στο κοινό μια οποιαδήποτε νέα εξέλιξη ή γεγονός ή και προϊόν. Υπάρχουν πολλές προσεγγίσεις ως προς τα δεδομένα που θα χρησιμοποιηθούν, τις πηγές άντλησής τους και τα χαρακτηριστικά που θα παρακολουθήσουμε και θα αξιολογήσουμε (metrics). Μια δημοφιλής τέτοια προσέγγιση, ειδικά των τελευταίων χρόνων όπου υπάρχουν πολλά ποιοτικά (και όχι απλώς αριθμητικά) δεδομένα διαθέσιμα μέσω των κοινωνικών δικτύων, είναι η συναισθηματική ανάλυση των κειμένων που παράγουν οι ίδιοι οι χρήστες/καταναλωτές. Πλέον υπάρχουν συνδρομητικές πλατφόρμες που αναλαμβάνουν παρόμοιες εργασίες, όπως υπάρχουν στην βιβλιογραφία πολλές επιτυχημένες προσπάθειες συναισθηματικής ανάλυσης με χρήση λεξικών προ-αξιολογημένων όρων.

Σκοπός της παρούσας εργασίας είναι να επιχειρήσει μια άλλη προσέγγιση συναισθηματικής ανάλυσης, χωρίς την ανάγκη χρήσης λεξικών αλλά με τεχνικές μη-επιβλεπόμενης μάθησης, συγκεκριμένα πάνω στην ελληνική γλώσσα για την οποία δεν υπάρχει αφθονία διαθέσιμων και εμπεριστατωμένων συναισθηματικών λεξικών. Στόχος δεν είναι να δημιουργηθεί μία ολοκληρωμένη λύση στην τελική της μορφή ή να εξετασθεί εις βάθος σε θεωρητικό επίπεδο ο μηχανισμός λειτουργίας των μοντέλων που θα χρησιμοποιηθούν, αλλά να διερευνηθεί η εφικτότητα και η πιθανή αποτελεσματικότητα μιας τέτοιας προσέγγισης για τους σκοπούς μιας γρήγορης αξιολόγησης της κοινής γνώμης, καθώς και να προταθούν τρόποι μελλοντικής βελτίωσής της.

Αρχικά, στο κεφάλαιο 2 θα αξιοποιηθεί η βιβλιογραφική ανασκόπηση για την εισαγωγή και επεξήγηση των όρων και των μεθοδολογιών που θα χρησιμοποιηθούν. Το 3ο κεφάλαιο παρουσιάζει αναλυτικά τα στάδια της υλοποίησης, από την συλλογή των δεδομένων μέσω του Twitter API και την προεπεξεργασία τους, έως την διερεύνηση της εφικτότητας επέκτασης των υπαρχόντων μοντέλων και την αξιοποίησή τους για τους σκοπούς της συναισθηματικής ανάλυσης. Το 4ο κεφάλαιο είναι αφιερωμένο στην συλλογή αποτελεσμάτων μέσω των προγραμματιστικών εργαλείων που παρουσιάστηκαν νωρίτερα και την αξιολόγησή τους:

- έγινε χρήση του κώδικα για την συλλογή 3 datasets διαφορετικής θεματικής
- επιλέχθηκαν τα κατάλληλα στάδια επεξεργασίας τους για αξιοποίηση από μοντέλα φυσικής γλώσσας (συγκριτικά με χρήση σε συνδυασμό με λεξικά)
- παρουσιάστηκε η εξαγωγή νέων λέξεων από τα επεξεργασμένα κείμενα για χρήση από τα μοντέλα NLI και καταδείχθηκε ότι η απλή ενσωμάτωσή τους σε αυτά δεν επιφέρει τα επιθυμητά αποτελέσματα

- τέλος, επαληθεύτηκε μέσω μικρών συνόλων προαξιολογημέων μηνυμάτων η αποτελεσματικότητα του μοντέλου στην αναγνώριση συναισθημάτων, παρουσιάστηκε μια επιπρόσθετη μέθοδος αξιολόγησης που μπορεί να οδηγήσει σε πιο εξισορροπημένα αποτελέσματα κατά την ανάλυση ενός συνόλου κειμένων μεικτών συναισθημάτων, και επιδείχθηκε η λειτουργία του κώδικα ανάλυσης πάνω στα αρχικά datasets.

Στο τελευταίο κεφάλαιο, παρουσιάζονται τα συμπεράσματά και προτείνονται σημεία στα οποία θα μπορούσε να επικεντρώσει ο μελλοντικός ερευνητής. Όπως αναφέρθηκε και παραπάνω, παρά τις πολλές βελτιώσεις που επιδέχεται αυτή η μέθοδος σε όλα τα στάδιά της (δεδομένου ότι έχει χρησιμοποιηθεί λίγο και κυρίως στα αγγλικά), ακόμα και σε αυτό το στάδιο μπορεί να επιφέρει ικανοποιητικά αποτελέσματα για τους σκοπούς μιας γρήγορης συναισθηματικής ανάλυσης. Η απόδοσή της πάνω σε προαξιολογημένα σύνολα προσεγγίζει κατά μέσο όρο την θεωρητική απόδοση του χρησιμοποιούμενου μοντέλου, ενώ τα αποτελέσματα που παράγονται από την αξιολόγηση των αρχικά συλλεγμένων datasets χαρακτηρίζονται λογικά.

2. Ανασκόπηση βιβλιογραφίας

Στο κεφάλαιο αυτό παρουσιάζεται μια ανασκόπηση της διεθνούς βιβλιογραφίας πάνω στα θέματα που εξετάζει η διπλωματική εργασία, όπως και μια επεξήγηση των κυριότερων όρων που θα χρησιμοποιηθούν στα επόμενα κεφάλαια. Γίνεται αναφορά γενικότερα στα κοινωνικά δίκτυα, ειδικότερα στο Twitter το οποίο επιλέχθηκε για την άντληση των αρχικών δεδομένων, περιγράφονται τα βασικότερα χαρακτηριστικά μέτρησης και αξιολόγησης επίδοσης (metrics) και επεξηγούνται όροι κλιμακούμενης εξειδίκευσης όπως η εξόρυξη δεδομένων (data mining), η συναισθηματική ανάλυση των δεδομένων αυτών (sentiment analysis), τα μοντέλα μηχανικής μάθησης (machine learning) τύπου μετασχηματιστή (transformers), και τεχνικές όπως τα Συμπεράσματα Φυσικής Γλώσσας (Natural Language Inference) και η ταξινόμηση μηδενικής βολής (zero-shot classification).

2.1. Κοινωνικά δίκτυα

Ένας γενικός ορισμός για τα κοινωνικά δίκτυα, θα μπορούσε να είναι “ένα σύνολο διαδικτυακών εφαρμογών που βασίζονται στα ιδεολογικά και τεχνολογικά θεμέλια του Web 2.0 και επιτρέπουν την δημιουργία και τον διαμοιρασμό περιεχομένου που δημιουργούν οι ίδιοι οι χρήστες” (Kaplan et al, 2010). Η δημιουργία και η μετέπειτα αλματώδης εξέλιξη των κοινωνικών δικτύων, εδράζεται στην επιθυμία του ανθρώπου για επικοινωνία σε συνδυασμό με την συνεχή εξέλιξη των ψηφιακών τεχνολογιών.

Το 2002, το LinkedIn ξεκίνησε ως μια ιστοσελίδα διασύνδεσης επαγγελματιών καριέρας. Το 2020 έφτασε να έχει περισσότερους από 670 εκατομμύρια χρήστες παγκοσμίως και να αποτελεί το κορυφαίο δίκτυο, τόσο για όσους αναζητούν εργασία όσο και για τους managers που αναζητούν κατάλληλους υποψηφίους. Δύο χρόνια αργότερα, το 2004, ο φοιτητής του Harvard Mark Zuckerberg δημιούργησε το Facebook - τώρα, διαθέτει περίπου 1,7 δισεκατομμύρια χρήστες. Το 2006, οι Jack Dorsey, Evan Williams και Biz Stone δημιούργησαν το Twitter ως μια πλατφόρμα μικρο-ιστολογίων (microblogging). Μέχρι πρόσφατα διέθετε περίπου 400 εκατομμύρια χρήστες, με το 23% των ενήλικων Αμερικανών πολιτών να διατηρούν λογαριασμό σε αυτό (Auxier et al, 2021). Το 2010 δημιουργήθηκε το Instagram από έναν άλλον φοιτητή, τον Kevin Systrom, ως ένας ιστότοπος διαμοιρασμού φωτογραφιών - δέκα χρόνια μετά, το Instagram έχει αγοραστεί από το Facebook και αριθμεί πάνω από 1,4 δισεκατομμύρια χρήστες. Μεγάλη ανάπτυξη γνώρισαν και γνωρίζουν πολλά ακόμα κοινωνικά δίκτυα των τελευταίων ετών, όπως το Pinterest, το Snapchat και πιο πρόσφατα το TikTok, δημιουργία της κινεζικής τεχνολογικής εταιρείας ByteDance που πλέον διαθέτει πάνω από 1 δισεκατομμύριο ενεργούς χρήστες.

Τα κοινωνικά δίκτυα ξεκίνησαν ως μια εμπειρία για τον σταθερό ή και τον φορητό υπολογιστή των χρηστών. Σύντομα όμως, η επέκταση των δικτύων κινητής τηλεφωνίας, η αύξηση της ταχύτητας πρόσβασης στο διαδίκτυο, και η εντυπωσιακή βελτίωση των δυνατοτήτων των κινητών τηλεφώνων (επεξεργαστική ισχύς, μεγαλύτερες οθόνες, πολύ ισχυρότερες κάμερες για φωτογραφίες και βίντεο) οδήγησε στην μετατόπιση της εμπειρίας αυτής προς τις φορητές συσκευές - επιτρέποντας στους χρήστες να έχουν πάντα “μαζί τους” τις κοινότητες στις οποίες συμμετέχουν. Έτσι, μέσα σε λιγότερο από μία γενεά, τα κοινωνικά δίκτυα έχουν εξελιχθεί από απλά εργαλεία άμεσου διαμοιρασμού ηλεκτρονικής πληροφορίας, σε εικονικούς χώρους συγκέντρωσης ανθρώπων - και χάρη στην πρόσβασή τους σε τεράστιους όγκους πληροφορίας, αναπόφευκτα μετατράπηκαν και σε κρίσιμης σημασίας διαύλους marketing και διαμόρφωσης της κοινής γνώμης (Maryville University, 2020).

2.2. Το Twitter

Το Twitter ξεκίνησε την λειτουργία του το 2006 και παραμένει έως και σήμερα ένα σημαντικό κοινωνικό δίκτυο και μία από τις επιδραστικότερες πλατφόρμες κοινωνικού και πολιτικού διαλόγου στο διαδίκτυο.

Η διάσταση της κοινωνικής δικτύωσης εδράζεται στην δυνατότητα να “ακολουθούμε” (follow) άλλους χρήστες - μια ενέργεια που υποδηλώνει ενδιαφέρον για τις δημοσιεύσεις των χρηστών αυτών και, μέσω της πιθανής αλλά όχι υποχρεωτικής ανταπόδοσής της, διαμορφώνει άτυπες

ιεραρχίες χρηστών βάσει του αριθμού και της αναλογίας ακολούθων/ακολουθούμενων (followers/following). Πέραν των προσωπικών, ιδιωτικών μηνυμάτων που συναντάμε στα περισσότερα κοινωνικά δίκτυα, στο Twitter υπάρχουν διάφοροι τύποι δημοσίων μηνυμάτων (tweets). Κάθε απλό μήνυμα είναι εξ'ορισμού δημόσιο και εκπέμπεται προς όλους τους χρήστες. Με την προσθήκη του συμβόλου @ και ενός ονόματος χρήστη, το μήνυμα εκλαμβάνεται ως απάντηση προς τον χρήστη εκείνο ή ως επισήμανση/αναφορά του (reply/mention), ενώ υπάρχει τόσο η δυνατότητα προώθησης/επαναμοιρασμού ενός μηνύματος (retweet ή εν συντομία RT) όσο και απλής επισήμανσής του ως "αγαπημένου" (favorite, το αντίστοιχο του Like άλλων κοινωνικών δικτύων). Τέλος, ο χρήστης μπορεί να επισημάνει θεματικά το μήνυμά του με την χρήση λέξεων/φράσεων-κλειδιών των οποίων προηγείται το σύμβολο της δίεσης (# - hashtags), διευκολύνοντας έτσι παράλληλα και την εύρεσή του από άλλους χρήστες που αναζητούν μηνύματα και συζητήσεις της ίδιας θεματικής (Jungherr, 2014).

Το πρώτο tweet δημοσιεύτηκε από τον ιδρυτή του Twitter Jack Dorsey στις 21 Μαρτίου του 2006 και χρειάστηκαν 3 χρόνια, 2 μήνες και μία ημέρα για να επιτευχθεί το ορόσημο του ενός δισεκατομμυρίου tweets στην πλατφόρμα. Το 2009, το Twitter εισήγαγε υπηρεσίες τοποθεσίας (location services) και την δυνατότητα της σήμανσης των μηνυμάτων με γεωγραφικό μήκος και πλάτος (geo-tagging) (Hamzah, 2018). Το Twitter αρχικά λειτούργησε ως μία υπηρεσία βασισμένη σε μηνύματα κινητής τηλεφωνίας SMS (short messaging service) και, για να διατηρήσει συμβατότητα με τον μέγιστο αριθμό 160 χαρακτήρων του προτύπου, επέτρεπε μηνύματα μήκους έως 140 χαρακτήρων, δεσμεύοντας τους υπόλοιπους 20 για το όνομα χρήστη και άλλες εντολές. Τον Σεπτέμβρη του 2017 όμως, ο αριθμός αυτός αυξήθηκε σε 280 με στόχο να επιτρέψει την καλύτερη έκφραση των χρηστών του κοινωνικού δικτύου, χωρίς όμως να χαθεί ο χαρακτήρας και η ιδιαιτερότητα των σύντομων μηνυμάτων (Rosen et al, 2017).

Όπως επισημάνθηκε, το Twitter είναι "ανοιχτό" υπό δύο έννοιες. Αφενός, όλες οι δημοσιεύσεις που αναρτώνται στο δίκτυο είναι από προεπιλογή δημόσιες και ορατές σε όλους, ενώ όλοι οι χρήστες μπορούν ανεμπόδιστα να αλληλεπιδράσουν με οποιονδήποτε άλλον. Αφετέρου, η προγραμματιστική διεπαφή (API) που διατίθεται παρέχει εύκολη πρόσβαση στα δεδομένα της πλατφόρμας. Τα παραπάνω έχουν καταστήσει το Twitter ένα εξαιρετικά δημοφιλές αντικείμενο μελέτης από ερευνητές και επιστήμονες μιας πληθώρας πεδίων και κλάδων - από την πολιτική και την διαφήμιση έως τις κοινωνικές επιστήμες και την διαχείριση κρίσεων - ενώ οι αλληλεπιδράσεις των χρηστών του προσεγγίζουν περισσότερο από οποιουδήποτε άλλου κοινωνικού δικτύου αυτό που αποκαλούμε "δημόσιο διάλογο" (Gaisbauer et al, 2021). Από την άλλη πλευρά, οι χρήστες του Twitter δεν είναι απαραίτητα αντιπροσωπευτικοί του κοινού - αντιθέτως, στατιστικά αποτελούν ένα έντονα ανομοιογενές δείγμα του γενικού πληθυσμού

(Mislove et al, 2011). Επιπλέον, επειδή α) η παραγωγή του περιεχομένου των κοινωνικών δικτύων είναι αποκεντρωμένη, β) η επιλογή για την κατανάλωσή του περιεχομένου αυτού (ποιους χρήστες και θεματικές θα ακολουθήσουμε) γίνεται σε επίπεδο ατόμου και γ) οι πλατφόρμες βελτιστοποιούνται για την προσφορά της καλύτερης δυνατής εμπειρίας στον χρήστη, υπάρχει ο προβληματισμός ότι τα κοινωνικά δίκτυα ενδεχομένως προάγουν την ιδεολογική πόλωση, καθώς οι χρήστες τείνουν να αναζητούν άλλους χρήστες με παρεμφερείς απόψεις (Hänska et al, 2019). Παρόλα αυτά, το Twitter προσφέρει ένα ανοιχτό δημόσιο πεδίο που προσφέρεται για συλλογή πληροφοριών, σχηματισμό απόψεων και επηρεασμό. Σε πολλές περιπτώσεις, δημοσιογράφοι και επαγγελματίες της επικοινωνίας ενσωματώνουν το Twitter στην καθημερινότητά τους, αναφέρονται στις απόψεις και τις τάσεις εντός της πλατφόρμας ως “κοινή γνώμη” και αντιμετωπίζουν αναρτήσεις με βαρύτητα παρόμοια μιας αναφοράς ειδησεογραφικού πρακτορείου (Gaisbauer et al, 2021).

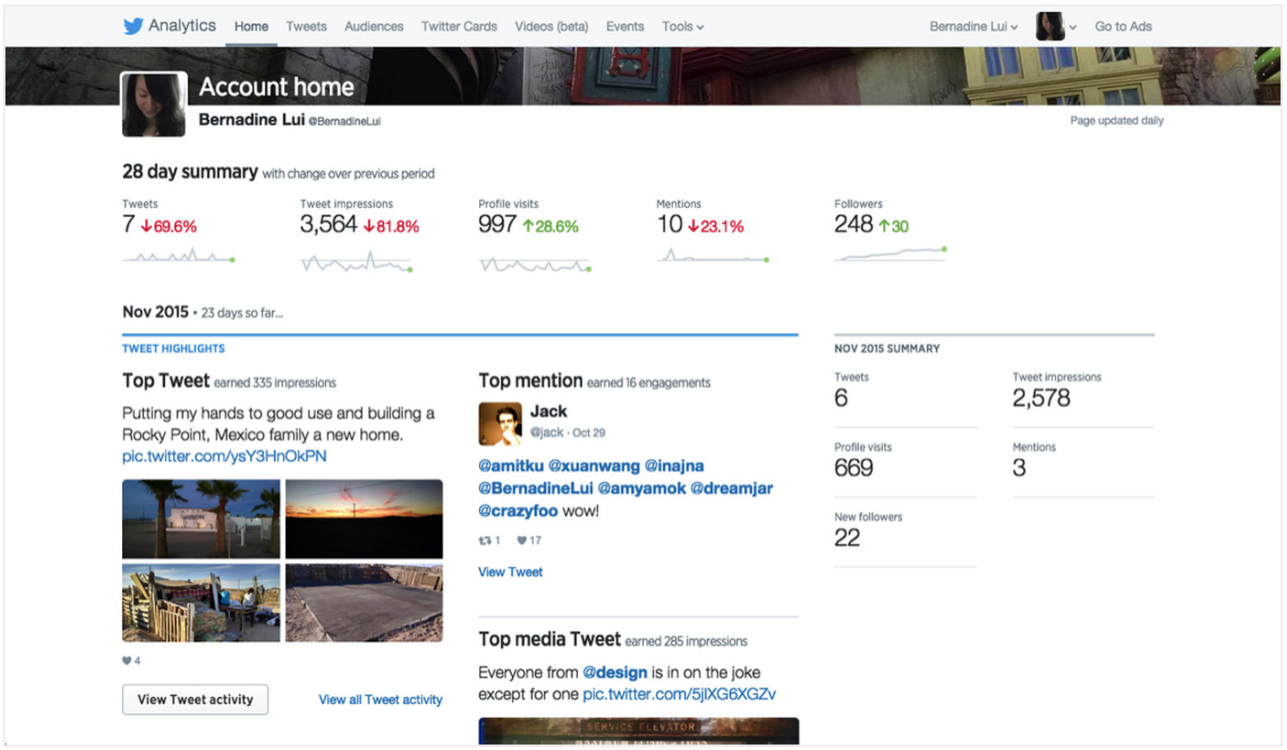
2.3. Metrics

Το Twitter, όπως τα περισσότερα κοινωνικά δίκτυα, συλλέγει, αποθηκεύει και επεξεργάζεται έναν πολύ μεγάλο όγκο πληροφοριών και στατιστικών στοιχείων (metrics) σχετικά με την δραστηριότητα των χρηστών του. Αυτά τα στοιχεία μπορούν να αξιοποιηθούν σε προσωπικό επίπεδο, σε επίπεδο οργανισμού, είτε ακόμα και σε πολύ μεγαλύτερη κλίμακα σε ερευνητικά σενάρια, και να αναλυθούν συνδυαστικά με άλλα δεδομένα για σκοπούς κατανόησης, εξαγωγής συμπερασμάτων και λήψης αποφάσεων (analytics).

Το ίδιο το Twitter παρέχει σε κάθε χρήστη πρόσβαση στα κυριότερα metrics που τον αφορούν, μέσω ειδικής σελίδας (https://analytics.twitter.com/). Μερικά από αυτά είναι τα εξής:

- Ο αριθμός των νέων μηνυμάτων που δημοσίευσε ο χρήστης (tweets)
- Ο αριθμός των ακολούθων του (followers)
- Οι συνολικές εμφανίσεις tweets του στην οθόνη άλλων χρηστών (tweet impressions)
- Οι επισκέψεις άλλων χρηστών στο προφίλ του (profile visits)
- Ο αριθμός επικλήσεων άλλων χρηστών στο username του χρήστη (mentions)

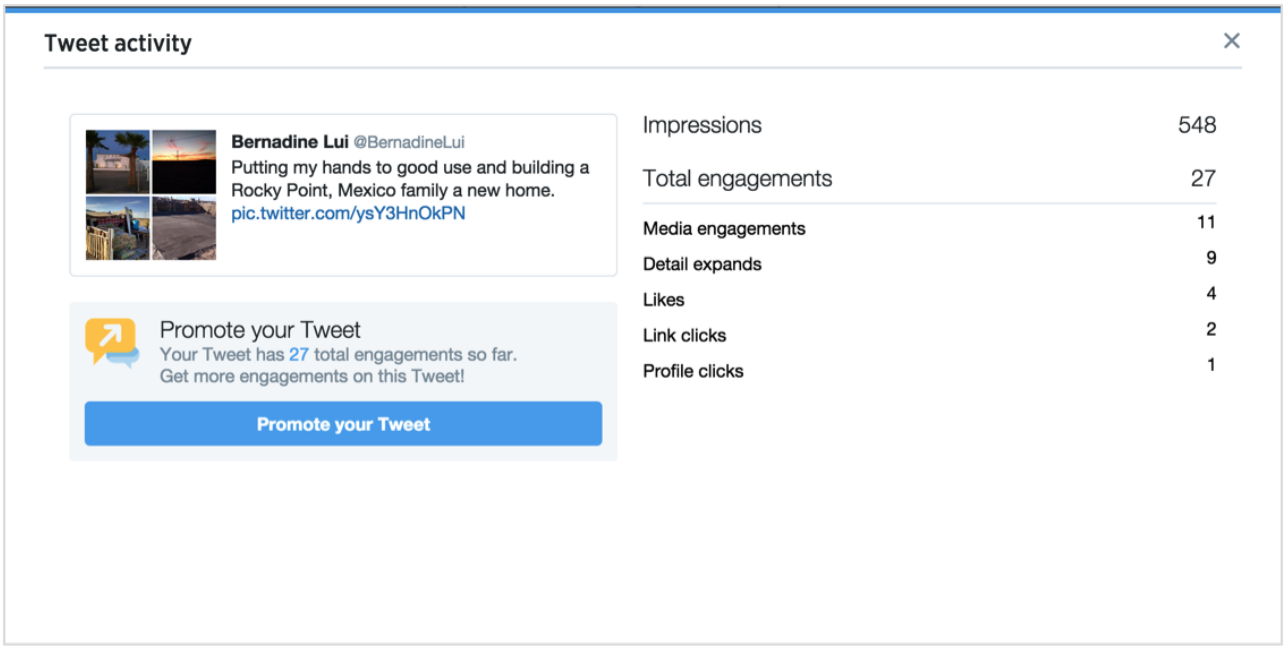
Για όλα τα παραπάνω metrics, εκτός από τον απόλυτο αριθμό ανά χρονική περίοδο (συνήθως μήνας) ή συνολικά από την δημιουργία του λογαριασμού χρήστη, διατίθεται και η ποσοστιαία μεταβολή από την προηγούμενη αντίστοιχη χρονική περίοδο.



Εικόνα 2.1 Metrics λογαριασμού χρήστη

Επιπλέον, παρέχονται επιπρόσθετες πληροφορίες ανά tweet, για τις αλληλεπιδράσεις που προκλήθηκαν μέσω της συγκεκριμένης δημοσίευσης, όπως:

- επισημάνσεις ως αγαπημένο (likes)
- αριθμός επαναμοιρασμών (retweets)
- αλληλεπιδράσεις με τα πολυμεσικά στοιχεία του μηνύματος, δηλαδή κλικς σε φωτογραφίες και βίντεο που τυχόν περιέχονται (media engagements)
- προβολές αναλυτικών λεπτομερειών του μηνύματος (detail expands)
- προβολές προσωπικού προφίλ του χρήστη, μέσω κλικς στο όνομα, το @username ή την φωτογραφία του μέσα από το συγκεκριμένο μήνυμα (profile clicks)
- νέοι ακόλουθοι που προέκυψαν μέσω του συγκεκριμένου μηνύματος (follows)



Εικόνα 2.2 Metrics μηνύματος

Τέλος, κυρίως για χρήστες διαφημιστικών υπηρεσιών του Twitter, καταγράφεται και διατίθεται μια σειρά metrics, επιπρόσθετα όσων έχουμε ήδη αναφέρει. Ενδεικτικά μερικά από αυτά:

- εγκαταστάσεις/ενεργοποιήσεις της εφαρμογής (για κλικς σε tweet ενσωματωμένο εντός ιστοσελίδας, μέσω κινητού τηλεφώνου ή tablet) (app install attempts/opens)
- ποσόστωση συνολικών αλληλεπιδράσεων ενός μηνύματος ως προς εμφανίσεις του (engagements to impressions rate)
- κλικς στην θεματική επισήμανση ενός tweet (hashtag clicks)
- κλικς σε υπερσυνδέσμους που περιέχονται σε ένα tweet (link clicks)
- κλικς στον υπερσύνδεσμο που αντιστοιχεί σε ενα tweet και οδηγεί απευθείας σε αυτό (permalink clicks)
- αριθμός φορών που το tweet προωθήθηκε μέσω email από χρήστες (shared via email)

Από τα παραπάνω, προκύπτει αβίαστα το συμπέρασμα ότι η “επίδοση” ενός tweet διακρίνεται από ανομοιογένεια και πολυπλοκότητα ως προς τους παράγοντες που την καθορίζουν. Ανάλογα με το πεδίο και το αντικείμενο της εκάστοτε έρευνας, μπορεί να χρειαστεί να εξεταστούν έντονα διαφοροποιημένοι μεταξύ τους παράγοντες, μεμονωμένα ή και συνδυαστικά - από το πόσα tweets, retweets και απαντήσεις το αφορούν, έως τα σχετικά hashtags, το πλήθος των εμπλεκομένων χρηστών και οι διάφορες υπερσυνδέσεις. Αυτή η εσωτερική ανομοιογένεια των

στατιστικών στοιχείων που καταγράφει το Twitter, η οποία οφείλεται στις πολλές διαφορετικές προσφερόμενες μορφές αλληλεπίδρασης και ανταλλαγής πληροφορίας, αποτελεί ερευνητική πρόκληση αλλά ταυτόχρονα και ευκαιρία. Για αυτόν τον λόγο, οι ερευνητές δίνουν όλο και μεγαλύτερη βαρύτητα στην ανάλυση τόσο του περιεχομένου του κειμένου των tweets μέσω της εξόρυξης δεδομένων (data mining), όσο και της συμπεριφοράς και του συναισθήματος των χρηστών του μέσου (sentiment analysis), ξεφεύγοντας από την απλή επεξεργασία αριθμητικών δεδομένων (Fang et al, 2020).

2.4. Data mining

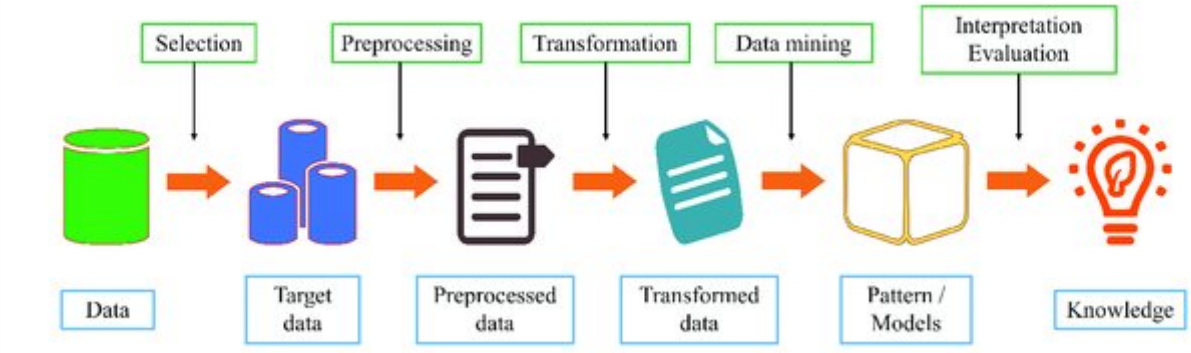
Ο γενικότερος όρος “εξόρυξη δεδομένων” αναφέρεται στην έρευνα, ανάπτυξη και εφαρμογή αυτοματοποιημένων μεθόδων ανίχνευσης προτύπων σε μεγάλες συλλογές δεδομένων, που θα ήταν πρακτικά πολύ δύσκολο έως αδύνατο να ανευρεθούν με άλλους τρόπους, ακριβώς λόγο του όγκου των δεδομένων μέσα στα οποία διαμορφώνονται (Romero et al, 2013).

Στην σύγχρονη εποχή, πρακτικά κάθε είδους αυτοματοποιημένο σύστημα παράγει δεδομένα, είτε για διαγνωστικούς σκοπούς, είτε για σκοπούς ανάλυσης. Αυτό αναπόφευκτα οδηγεί στην δημιουργία μιας υπερπληθώρας πληροφοριών, τις οποίες οι αναλυτές αξιοποιούν για την λήψη καλύτερων αποφάσεων και την βελτίωση των διαδικασιών. Οι βασικότερες πηγές παραγωγής τέτοιων δεδομένων, καθώς και κάποιοι από τους τομείς στους οποίους τα αξιοποιεί η επιστήμη της εξόρυξης δεδομένων, αναφέρονται παρακάτω:

- *Το διαδίκτυο/παγκόσμιος ιστός:* ηλεκτρονικό εμπόριο, συστήματα προτάσεων (βάσει συναλλαγών, προφίλ χρηστών, λέξεων-κλειδιών, χαρακτηριστικών προϊόντων), ομαδοποίηση πελατών για στοχευμένη διαφήμιση
- *Οι οικονομικές συναλλαγές:* πρόβλεψη δανείων, ανάλυση χρήσης πιστωτικών καρτών, ανίχνευση/πρόληψη απάτης και άλλων οικονομικών εγκλημάτων
- *Οι αλληλεπιδράσεις χρηστών* (π.χ. αρχεία κλήσεων στις τηλεπικοινωνίες ή καταναλωτική συμπεριφορά στα λιανικά καταστήματα): διαφημιστικές εκστρατείες, διατήρηση πελατών, βέλτιστη κατανομή πόρων και υποδομών, βελτιστοποίηση προσφερόμενων πακέτων/προγραμμάτων/υπηρεσιών
- *Αισθητήρες και IoT (Internet of Things)*: αναγνώριση και ανάλυση συμπτωμάτων και ασθενειών, αναγνώριση σχετιζόμενων παραγόντων και πιθανών αιτίων, επιλογή αποτελεσματικότερων θεραπειών, πρόβλεψη και αποτροπή εξάπλωσης ασθενειών/πανδημιών (Gupta et al, 2020)

Όλα τα παραπάνω πρωτογενή δεδομένα, τις περισσότερες φορές χαρακτηρίζονται από ανομοιογένεια και είναι αδόμητα, ενώ μπορεί να μην βρίσκονται καν σε μορφή που να επιτρέπει την περαιτέρω αυτοματοποιημένη επεξεργασία τους. Για αυτόν τον λόγο, οι επιστήμονες της εξόρυξης χρησιμοποιούν μια συγκεκριμένη διαδικασία "προεπεξεργασίας", κατά την οποία τα δεδομένα που συλλέχθηκαν "καθαρίζονται" και αποκτούν μια προτυποποιημένη μορφή. Η προεπεξεργασία αποτελεί ίσως και το πιο επίπονο τμήμα της διαδικασίας εξόρυξης - ακριβώς λόγω της μεγάλης ποικιλίας, τόσο στα προβλήματα που καλείται να αντιμετωπίσει, όσο και στις μορφές δεδομένων που απαιτείται να διαχειριστεί. Η όλη διαδικασία θυμίζει έντονα αυτήν της πραγματικής εξόρυξης ενός πρωτογενούς μεταλλεύματος και της σταδιακής μετατροπής του σε ένα τελικό προϊόν, από την οποία προήλθε και ο όρος "εξόρυξη δεδομένων" (Aggarwal, 2015).

Μια επιπλέον πρόκληση που εμφανίστηκε τα τελευταία χρόνια, αφορά τον τεράστιο όγκο δεδομένων που παράγονται ασταμάτητα από πηγές όπως τα κοινωνικά δίκτυα, και έχει οδηγήσει στην στροφή του ερευνητικού ενδιαφέροντος προς την κατεύθυνση των ροών δεδομένων (data streams). Τόσο μεγάλες ροές είναι πολύ δύσκολο να αποθηκευτούν πριν την επεξεργασία χωρίς την χρήση σημαντικών πόρων και υποδομών αποθήκευσης. Σε τέτοιες περιπτώσεις, όπου η αποθήκευση δεν είναι εφικτή για τους σκοπούς της ανάλυσης, η επεξεργασία των δεδομένων πρέπει να γίνει σε πραγματικό χρόνο.



Εικόνα 2.3 Στάδια εξόρυξης δεδομένων (Yang et al, 2020)

Μία τυπική διαδικασία εξόρυξης δεδομένων περιλαμβάνει τις εξής φάσεις:

I. *Συλλογή δεδομένων (data collection).* Το στάδιο της συλλογής μπορεί να περιλαμβάνει την χρήση εξειδικευμένου υλικού/συσκευών όπως αισθητήρες σε μια γραμμή παραγωγής ή σε ένα νοσοκομείο, εξειδικευμένου λογισμικού όπως για την ανίχνευση και σάρωση ιστοσελίδων του διαδικτύου ή εγγράφων σε ένα τοπικό εταιρικό δίκτυο, ή ακόμα και ανθρώπινη χειρονακτική εργασία - για παράδειγμα την συλλογή ερωτηματολογίων μέσω μιας έρευνας στον δρόμο. Παρόλο που το στάδιο αυτό είναι

ιδιαίτερα εξειδικευμένο αναλόγως του κάθε προβλήματος και συνήθως έξω από το άμεσο αντικείμενο των αναλυτών της εξόρυξης δεδομένων, είναι ιδιαίτερης σημασίας αφού οι καλές επιλογές εδώ (ποια δεδομένα θα συλλεχθούν και ποια όχι, σε τι μορφή) θα διευκολύνουν σημαντικά τα επόμενα στάδια της διαδικασίας. Σε κάθε περίπτωση, τα αποτελέσματα αυτής της φάσης καταλήγουν σε βάσεις ή αποθήκες δεδομένων (databases / data warehouses) ώστε να ακολουθήσει η επεξεργασία τους.

II. *Επιλογή-εξαγωγή χαρακτηριστικών και καθαρισμός δεδομένων (feature selection-extraction & data cleaning):* μετά την συλλογή τους, τα δεδομένα συνήθως δεν βρίσκονται σε μορφή κατάλληλη για επεξεργασία, οπότε είναι απαραίτητη η μετατροπή τους σε τύπους "φιλικούς" προς τους αλγόριθμους εξόρυξης. Τα δεδομένα που συλλέχθηκαν περιλαμβάνουν πολλές διαφορετικές καταγεγραμμένες ιδιότητες και παραμέτρους, οι οποίες αναφέρονται ως χαρακτηριστικά (features) ή γνωρίσματα (attributes) ή διαστάσεις (dimensions). Η διαδικασία επιλογής και εξαγωγής αποσκοπεί στην αναγνώριση εκείνων των χαρακτηριστικών των δεδομένων που θα είναι χρήσιμα για τους σκοπούς της εξόρυξης - αντί να προσθέτουν αχρείαστες διαστάσεις που δεν συμβάλλουν στο τελικό αποτέλεσμα - και επιπλέον στην δημιουργία πιθανών νέων συνδυαστικών χαρακτηριστικών προς αντικατάσταση των αρχικών, που θα οδηγήσει στην μείωση της πολυπλοκότητας του συστήματος (Aparna et al, 2016). Οι παραπάνω διαδικασίες συνήθως διενεργούνται παράλληλα με τον καθαρισμό των δεδομένων, όπου λανθασμένα ή ελλειπή τμήματα των δεδομένων, διορθώνονται ή συμπληρώνονται κατ' εκτίμηση, ενώ κάποια πιθανώς μη χρήσιμα ακόμα και διαγράφονται. Το τελικό αποτέλεσμα αυτής της φάσης είναι ένα σύνολο δεδομένων (dataset) με σωστή δομή, το οποίο μπορεί να γίνει αντικείμενο αποδοτικής επεξεργασίας από ένα υπολογιστικό σύστημα.

III. *Αναλυτική επεξεργασία και αλγόριθμοι:* Το τελευταίο στάδιο της διαδικασίας εξόρυξης, είναι ο σχεδιασμός αποδοτικών μεθόδων ανάλυσης των επεξεργασμένων δεδομένων. Για να γίνει αυτό, τα προβλήματα ανάλυσης έχουν κωδικοποιηθεί σε 4 βασικά είδη, τα οποία χρησιμοποιούνται και ως θεμέλιοι λίθοι κάθε τέτοιου σχεδιασμού.

    - *εξόρυξης μοτίβων συσχετίσεων* (association pattern mining): αφορά στην ανίχνευση μοτίβων ή συνυπάρξεων μέσα σε δεδομένα, και συσχετίσεων τύπου if-then (για παράδειγμα, αν κάποιος αγοράζει ένα συγκεκριμένο προϊόν, ποιο άλλο προϊόν προτιμάει με πιθανότητα πάνω από ένα όριο).

- *συσταδοποίησης* (clustering): αφορά στον διαχωρισμό ενός συνόλου δεδομένων ή αντικειμένων σε διακριτές ομάδες (συστάδες) με παρεμφερή χαρακτηριστικά (για παράδειγμα, ομαδοποίηση πελατών βάσει πολλαπλών κριτηρίων για αποτελεσματικότερη στόχευση με κατάλληλες προσφορές)

- *ανίχνευσης ακραίων στοιχείων* (outlier detection): αφορά στην ανίχνευση τιμών κάποιων χαρακτηριστικών που διαφέρουν από το αναμενόμενο τόσο αισθητά ώστε να προκαλέσουν υπόνοιες ύπαρξης προβλήματος (για παράδειγμα, ασυνήθιστες/υποπτες κινήσεις πιστωτικών καρτών, συμβάντα αισθητήρων, ανίχνευση ιατρικών καταστάσεων)

- *ταξινόμησης δεδομένων* (data classification): αφορά στην περίπτωση που θέλουμε να εξετάσουμε ένα πρόβλημα εξόρυξης δεδομένων και να αξιολογήσουμε τις τιμές των χαρακτηριστικών του, υπό το πρίσμα ενός συγκεκριμένου χαρακτηριστικού αναφοράς του dataset - που θα προσδιορίζεται ως “ετικέτα κλάσης” (class label). Σε αυτά τα προβλήματα, το σύστημα “εκπαιδεύεται” να αναζητά τις σχέσεις των χαρακτηριστικών του dataset με το χαρακτηριστικό αναφοράς σε υπάρχοντα δεδομένα (training data), ώστε μετά να μπορεί να εκτιμά την τιμή του χαρακτηριστικού αναφοράς όταν αυτό απουσιάζει σε ένα μεταγενέστερο dataset (για παράδειγμα πρόβλεψη εκλογικής συμπεριφοράς, ή αξιοπιστίας ενός επίδοξου δανειολήπτη βάσει δημογραφικών και άλλων χαρακτηριστικών) (Aggarwal, 2015).

2.5. Sentiment analysis

Η Ανάλυση Συναισθήματος αποτελεί μια λειτουργία της Εξόρυξης Δεδομένων και αφορά στην μελέτη με υπολογιστικά μέσα των ανθρώπινων απόψεων, στάσεων, εκτιμήσεων και συναισθημάτων ως προς κάποια οντότητα, όπως αυτά αποτυπώνονται στον γραπτό λόγο - η οντότητα αυτή μπορεί να αναπαριστά άτομα, γεγονότα ή και θεματικές ενότητες (Medhat et al, 2014). Η τεχνική αυτή μπορεί να βρει εφαρμογή σε πεδία όπως οι κριτικές προϊόντων και υπηρεσιών, η ειδησεογραφία, η πολιτική ή ακόμα και οι χρηματαγορές. Στο πεδίο της πολιτικής για παράδειγμα, μπορούμε να διερευνήσουμε την στάση του κοινού απέναντι σε κόμματα και υποψηφίους, ενώ έχουν γίνει και προσπάθειες πρόβλεψης εκλογικών αποτελεσμάτων μέσα από την ανάλυση πολιτικών δημοσιεύσεων (Ramteke et al, 2016). Τα κοινωνικά δίκτυα είναι επίσης μια πάρα πολύ καλή πηγή “πρώτης ύλης” για την Ανάλυση Συναισθήματος, λόγω του ελεύθερου και μαζικού τρόπου με τον οποίο οι χρήστες εκφράζουν σε αυτά τις απόψεις τους πάνω σε συγκεκριμένα θέματα. Για αυτόν τον λόγο, η διεύρυνση της σημασίας της Ανάλυσης

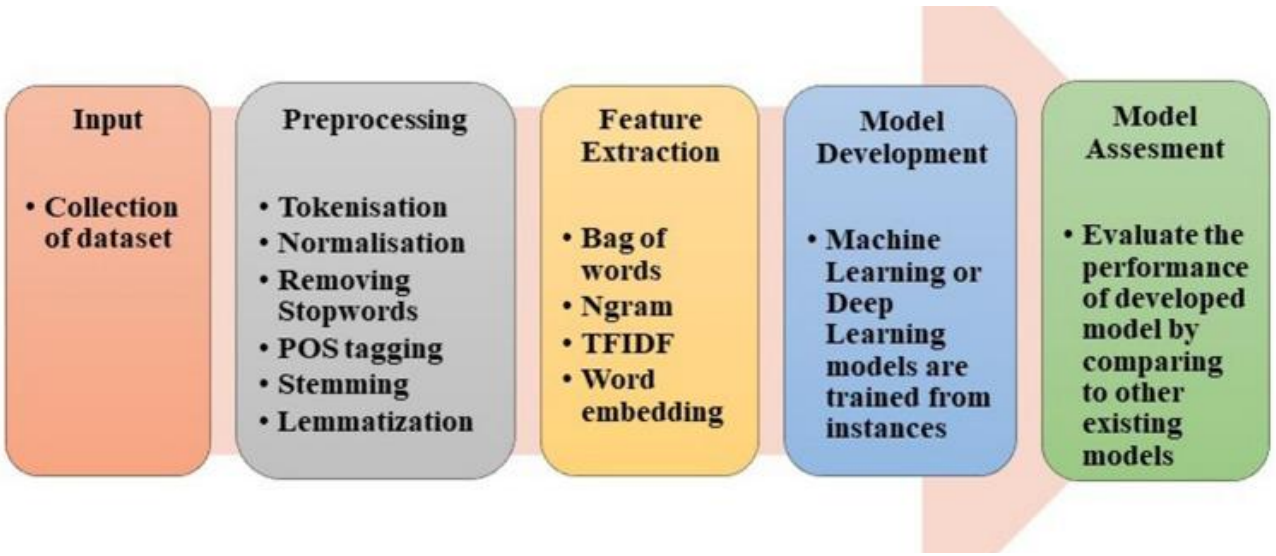
Συναισθήματος συμβαδίζει με την ανάπτυξη των πάσης φύσεως κοινωνικών δικτύων και έχει επεκταθεί εκτός των στενών συνόρων της πληροφορικής, στην σφαίρα των κοινωνικών επιστημών και των επιστημών διοίκησης (Cardie, 2014).

Η Ανάλυση Συναισθήματος θα μπορούσαμε να πούμε ότι είναι μια επιμέρους κατεύθυνση ή τεχνική του ευρύτερου τομέα της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing - NLP), που με την σειρά της αποτελεί τομέα της Τεχνητής Νοημοσύνης - με αντικείμενο την ανάγνωση, κατανόηση και εξαγωγή νοήματος από τις ανθρώπινες γλώσσες. Εκτός από την Ανάλυση Συναισθηματος, η Επεξεργασία Φυσικής Γλώσσας αξιοποιείται και σε άλλες διαδικασίες ανάλυσης όπως:

- στην Εξόρυξη Γνώμης (Opinion Mining), που αποτελεί μια πιο λεπτομερή και αναλυτική εκδοχή ανάλυσης συναισθημάτων, ως προς συγκεκριμένες πτυχές ενός θέματος.
- στην εξαγωγή φράσεων-κλειδιών, για την γρήγορη ανίχνευση των βασικών θεμάτων σε ένα απόσπασμα κειμένου.
- στην ανίχνευση Προσωπικά Αναγνωρίσιμων Στοιχείων (Personally identifiable information - PII) στο εσωτερικό μεγάλων εγγράφων.
- στην ανίχνευση της βασική γλώσσας ενός εγγράφου, μεταξύ πολλών διαφορετικών γλωσσών, διαλέκτων και ιδιωμάτων (Forvis, 2020).

Η Ανάλυση Συναισθήματος πραγματεύεται την αναγνώριση της πολικότητας σε ένα κομμάτι γραπτού λόγου (από μία φράση έως ένα ολόκληρο έγγραφο), και συγκεκριμένα αν η διάθεση απέναντι στην εξεταζόμενη οντότητα χαρακτηρίζεται ως θετική, αρνητική ή ουδέτερη. Πολλές φορές όμως, οι ανάγκες κατανόησης απαιτούν λεπτομερέστερη ανάλυση, οπότε και χρησιμοποιούνται κλίμακες περισσότερων σημείων (για παράδειγμα 5 σημείων όπως “έντονη διαφωνία, διαφωνία, ουδετερότητα, συμφωνία, έντονη συμφωνία”) ή κλίμακες πιο συγκεκριμένης κατηγοριοποίησης του συναισθήματος. Τις τελευταίες δεκαετίες έχουν προταθεί πολλά διαφορετικά “συναισθηματικά μοντέλα” αλλά αυτό που έχει επικρατήσει και χρησιμοποιείται περισσότερο μεταξύ των ερευνητών είναι το μοντέλο 6 συναισθηματικών καταστάσεων του Ekman (θυμός, αποστροφή, φόβος, ευχαρίστηση, λύπη, έκπληξη) (Ekman, 1992).

Η διαδικασία της Ανάλυσης Συναισθήματος μπορεί να περιλαμβάνει διάφορα στάδια, με μικρές διαφοροποιήσεις αναλόγως του επιλεγμένου τρόπου υλοποίησης. Στο παρακάτω σχήμα παρουσιάζεται μια ενδεικτική ακολουθία και αμέσως μετά γίνεται μία συνοπτική περιγραφή των σταδίων.



Εικόνα 2.4 Στάδια ανάλυσης συναισθήματος (Nandwani et al, 2021)

I. **Συλλογή των δεδομένων.** Τα σύνολα δεδομένων (datasets) που χρησιμοποιούνται στην Ανάλυση Συναισθήματος προέρχονταν στις αρχές της ψηφιακής εποχής από πηγές όπως ειδησεογραφικούς ιστότοπους, κριτικές, ιστολόγια (blogs), αλληλογραφίες. Τα τελευταία όμως χρόνια, πολλές έρευνες τείνουν να χρησιμοποιούν δεδομένα από πλατφόρμες κοινωνικής δικτύωσης όπως το Twitter, το Facebook και το Youtube. Σε κάθε περίπτωση, όλα αυτά τα δεδομένα είναι κυρίως αδόμητα, οπότε κρίνεται απαραίτητο να περάσουν από μια διαδικασία επεξεργασίας που θα τους δώσει την κατάλληλη δομή ώστε να μπορούν να αξιοποιηθούν υπολογιστικά με έναν αποδοτικό τρόπο.

II. **Προεπεξεργασία.** Η προεπεξεργασία αποτελεί ένα κρίσιμο στάδιο αφού καθαρίζει και δομεί το dataset και έτσι καθορίζει την ποιότητα των δεδομένων πάνω στα οποία θα βασιστεί η ανάλυση. Ταυτόχρονα, θα πρέπει να δοθεί προσοχή ώστε να ελαχιστοποιηθεί η απώλεια τυχόν χρήσιμων δεδομένων, που πολλές φορές είναι αναπόφευκτη κατά την προεπεξεργασία (Bhaskar et al, 2015). Το στάδιο αυτό μπορεί, αναλόγως του αντικειμένου της μελέτης, να περιλαμβάνει αρκετές επιμέρους υποδιεργασίες όπως:

    - την επιμεροποίηση μεγαλύτερων κομματιών κειμένου - ολόκληρων εγγράφων, παραγράφων ή ακόμα και προτάσεων - σε μικρές ομάδες λέξεων (γνωστές ως tokens - tokenization). Ταυτόχρονα διορθώνεται και η ορθογραφία των λέξεων ώστε να επιτευχθεί η μέγιστη δυνατή ομοιομορφία των δεδομένων.

- την αφαίρεση λέξεων που δεν συμβάλλουν στην ανίχνευση και την ανάλυση συναισθημάτων, όπως για παράδειγμα άρθρα (ο, η, το), προθέσεις (σε, ως, προς, για, σαν) ή και κάποια ρήματα (είναι, έχω).

- την επισήμανση λέξεων ως προς το μέρος του λόγου που αποτελουν (POS tagging, part-of-speech) μέσα σε μία πρόταση. Η διεργασία αυτή διευκολύνει την αναγνώριση και τον διαχωρισμό μεταξύ συναισθημάτων (που συνήθως εκφράζονται μέσω επιθέτων) και θεμάτων (που συνήθως περιγράφονται μέσω ουσιαστικών) (Sun et al, 2017).

- την αφαίρεση των καταλήξεων (stemming) και την λημματοποίηση (lemmatization) των λέξεων. Οι δύο διεργασίες είναι παρόμοιες μεταξύ τους, αλληλοσυμπληρούμενες και αποσκοπούν στο να μετατρέψουν παρεμφερείς λέξεις σε μια “ριζική”, προτυποποιημένη μορφή, έτσι όπως δηλαδή θα αποτυπωνόταν σε ενα λεξικό (για παράδειγμα, ενικός, ενεστώτας, ονομαστική πτώση, χωρίς υποκοριστικά κτλ.) (Symeonidis et al, 2018).

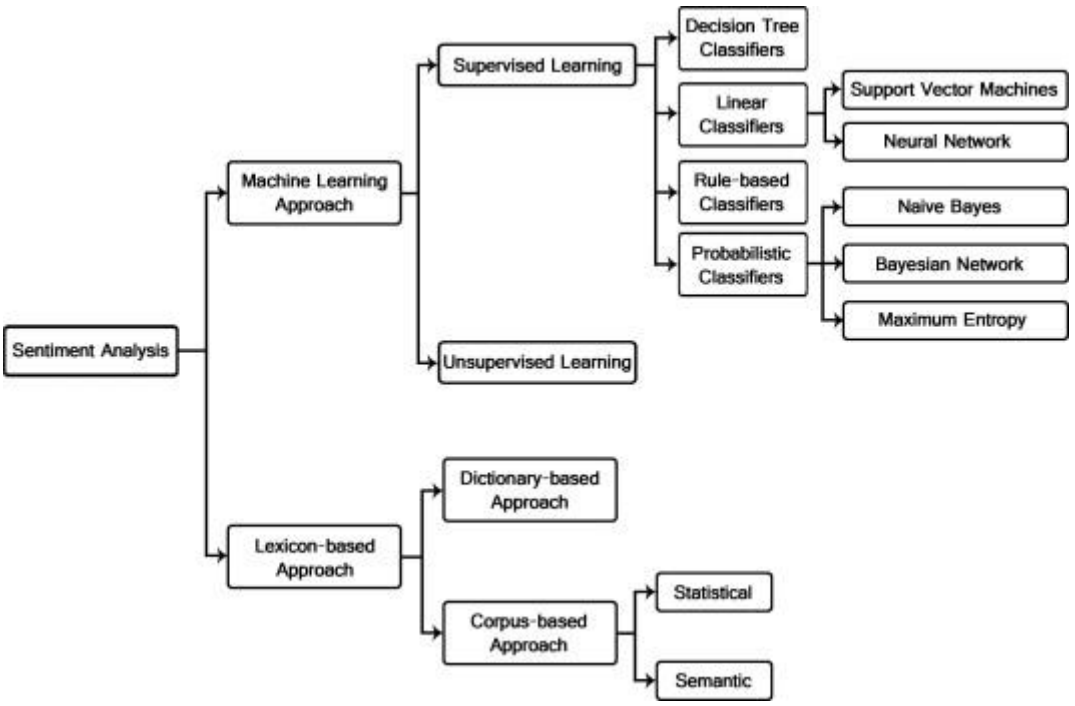
III. **Εξαγωγή χαρακτηριστικών**. Στο στάδιο αυτό γίνεται η αναγνώριση των θεμάτων (aspects) που σχολιάζονται στο κείμενο ή αλλιώς των χαρακτηριστικών της οντότητας για την οποία διενεργούμε την Ανάλυση Συναισθήματος. Οι υπολογιστές αντιλαμβάνονται το κείμενο με αριθμητικούς όρους. Τις περισσότερες φορές, η διαδικασία περιλαμβάνει την δημιουργία ενός χάρτη/πίνακα χαρακτηριστικών, όπου τα θέματα προκύπτουν από την συχνότητα εμφάνισης συγκεκριμένων ουσιαστικών στο επεξεργασμένο dataset - ειδικά όταν αυτά τα ουσιαστικά συνοδεύονται από επίθετα τα οποία, όπως έχουμε ήδη αναφέρει, μεταφέρουν συναίσθημα (Siqueira et al, 2010).

IV. **Ανάπτυξη του μοντέλου**. Σε γενικές γραμμές, όπως φαίνεται και στο σχήμα που ακολουθεί, υπάρχουν δύο μεθοδολογίες για την ανάπτυξη των μοντέλων ανάλυσης συναισθημάτων - αυτή της χρήσης λίστας λέξεων (lexicon) και αυτή της χρήσης μηχανικής μάθησης (machine learning).

- Lexicon: η μέθοδος αυτή αξιοποιεί λίστες λέξεων και αποδίδει θετικές ή αρνητικές βαρύτητες σε όσες λέξεις του αξιολογούμενου κειμένου φέρουν συναισθηματικό περιεχόμενο. Κατόπιν, το άθροισμα αυτών των θετικών και αρνητικών τιμών χρησιμοποιείται για τον υπολογισμό του συνολικού συναισθήματος μιας πρότασης έως και ενός ολόκληρου εγγράφου. Με την σειρά της, η μέθοδος λίστας λέξεων διακρίνει δύο τύπους προσεγγίσεων - την προσέγγιση *βάσει λεξικού* (dictionary-based) και την προσέγγιση *βάσει σώματος*

κειμένων (corpus-based). Η κύρια διαφοροποίησή του είναι πως στην πρώτη περίπτωση τηρείται ένα συστηματικά οργανωμένο λεξικό με λέξεις της ίδιας γλώσσας, ενώ στην δεύτερη χρησιμοποιούνται τυχαία αποσπάσματα κειμένου και στατιστικά εμφάνισης λέξεων ή συντακτικών patterns (Darwich et al, 2019). Η corpus-based προσέγγιση προσφέρει καλύτερα αποτελέσματα όταν η Ανάλυση Συναισθήματος πρέπει να γίνει επί ενός συγκεκριμένου θεματικού τομέα (οπότε και οι λέξεις αξιολογούνται συναισθηματικά υπό ένα συγκεκριμένο πρίσμα), αλλά τείνει να μην έχει εξίσου καλά αποτελέσματα σε πιο γενικές περιπτώσεις.

- Machine-learning: η γενική αρχή της μεθόδου αυτής είναι ότι το σύνολο των προεπεξεργασμένων δεδομένων των προηγουμένων σταδίων χωρίζεται σε δύο τμήματα - τα δεδομένα εκπαίδευσης (training dataset) που περιέχουν γνωστές πληροφορίες για χαρακτηριστικά της εξεταζόμενης οντότητας βάσει των οποίων θα εκπαιδευτεί το μοντέλο μας, και τα δεδομένα ελέγχου (testing dataset) βάσει των οποίων οι ερευνητές αξιολογούν πόσο επιτυχημένη ήταν η εκπαίδευση του μοντέλου. Και αυτή η μέθοδος διακρίνει δύο προσεγγίσεις - τα *επιβλεπόμενα* και τα *μη-επιβλεπόμενα* μοντέλα. Τα επιβλεπόμενα μοντέλα προϋποθέτουν την χρήση δεδομένων που έχουν ήδη επισημανθεί και κατηγοριοποιηθεί ως προς κάποια χαρακτηριστικά τους (κάτι που είναι συχνά χρονοβόρο, ενώ κάποιες φορές ίσως και ανέφικτο), ώστε να εκπαιδευτούν να αναγνωρίζουν το πώς κάθε λέξη επηρεάζει το συναισθηματικό αποτέλεσμα. Αντίθετα, τα μη-επιβλεπόμενα μοντέλα δεν χρησιμοποιούν προϋπάρχουσες πληροφορίες ως προς την επίδραση των εκάστοτε χαρακτηριστικών - απλώς δέχονται τα πρωτογενή δεδομένα και τα αναλύουν με σκοπό να αναγνωρίσουν πρότυπα και δομές χωρίς ανθρώπινη παρέμβαση (Wójcik, 2019; Delua, 2021).



Εικόνα 2.5 Μεθοδολογίες ανάλυσης συναισθήματος (Medhat et al, 2014)

V. **Αξιολόγηση**. Η αξιολόγηση κάθε μοντέλου προϋποθέτει την ύπαρξη συγκεκριμένων δεικτών μετρήσεων (metrics) που θα ποσοτικοποιούν την απόδοσή του. Πολλές φορές αξιοποιούνται οι λεγόμενοι πίνακες σύγχυσης (confusion matrix), οι οποίοι αποτυπώνουν γραφικά τις σωστές και λανθασμένες εκτιμήσεις/προβλέψεις ενός μοντέλου, από τις οποίες προκύπτουν επιπρόσθετοι δείκτες αξιολόγησης (accuracy, precision, F1 score, recall κτλ).

Ως επίλογο, πρέπει να αναφερθούμε και σε μερικές δεδομένες προκλήσεις που αντιμετωπίζουν οι ερευνητές της Ανάλυσης Συναισθήματος, με την μεγαλύτερη να αποτελεί η έλλειψη πόρων. Η συλλογή των δεδομένων αυτή καθαυτή δεν είναι δύσκολη, αλλά η ανθρώπινη επεξεργασία τους για την αναθεση συναισθηματικών βαρών είναι χρονοβόρα και όχι πάντα ακριβής ή αξιόπιστη, ενώ τα περισσότερα έτοιμα datasets και λεξικά είναι στην Αγγλική γλώσσα ή έντονα θεματικά - κάτι που δεν επιτρέπει την γενικευμένη αξιοποίησή τους. Η δεύτερη μεγάλη πρόκληση έγκειται στην ευρεία χρήση εναλλακτικών τρόπων γραφής, ειδικά στα κοινωνικά δίκτυα, όπως η αργκό, οι συντομευμένες λέξεις και τα εικονίδια έκφρασης (emoticons-emojis) - μορφές που δεν μπορούν να αξιοποιηθούν εύκολα στην αυτοματοποιημένη Ανάλυση Συναισθήματος. Τέλος, υπάρχουν και οι αντικειμενικές δυσκολίες που προκύπτουν από τον τρόπο εκφοράς του λόγου όπως η δύσκολα ανιχνεύσιμη χρήση σαρκασμού/ειρωνείας, ή η έκφραση πολλαπλών συναισθημάτων μέσα στην ίδια πρόταση (Nandwani et al, 2021).

2.6. Transformers

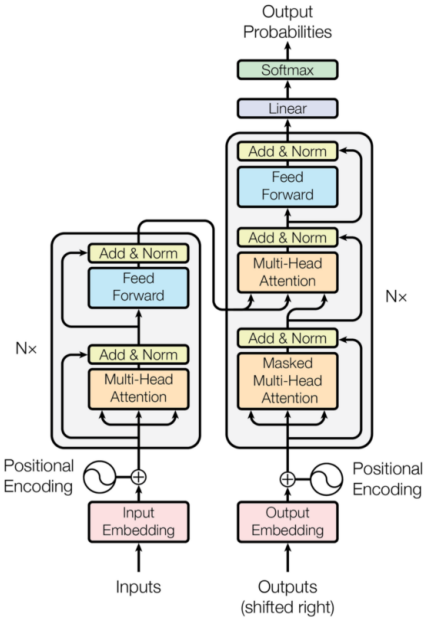
Εδώ και πολλά χρόνια, οι επιστήμες των δεδομένων επιδιώκουν την αποτελεσματικότερη δυνατή εξαγωγή νοήματος από κείμενα, με σκοπό την υποβοήθηση γλωσσικών εργασιών όπως οι αυτοματοποιημένες μεταφράσεις, η αυτόματη παραγωγή κειμένων ή περιλήψεών τους, η απάντηση ερωτήσεων, η διόρθωση κειμένων, η ανίχνευση συναισθημάτων και άλλα (Couto, 2015). Ο τομέας της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing - NLP) ασχολείται ειδικά με την ανεύρεση λύσεων στα παραπάνω θέματα.

Τα πρώτα μοντέλα που αναπτύχθηκαν για σκοπούς εξαγωγής νοήματος για υποβοήθηση γλωσσικών εργασιών, ήταν μεν αποτελεσματικά αλλά περιορίζονταν στην μία και μόνη λειτουργία για την οποία είχαν δημιουργηθεί και εκπαιδευτεί. Το αποτέλεσμα ήταν να απαιτείται ουσιαστικά η δημιουργία μιας νέας λύσης κάθε φορά που άλλαζαν παράμετροι εφαρμογής μιας υπάρχουσας (για παράδειγμα, η γλώσσα), ή πολύ περισσότερο όταν προέκυπτε ένας διαφορετικός σκοπός. Μία σημαντική εξέλιξη στον τομέα του NLP ήταν τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNN), τα οποία χρησιμοποιήθηκαν εντατικά για αρκετά χρόνια. Αλλά και αυτά δεν ήταν τέλεια - η δομή τους δεν επέτρεπε την παράλληλη επεξεργασία των δεδομένων που λάμβαναν, στερώντας τους ταχύτητα, ένω δεν απέδιδαν καλά σε περιπτώσεις κειμένων/εισόδων πολύ μεγάλου μήκους, λόγω του φαινομένου της διαδοχικά όλο και μικρότερης βελτίωσης της λύσης του μοντέλου σε κάθε επανάληψη (“vanishing gradient”) (Gillioz et al, 2020).

Το 2017, προτάθηκε μία νέα λύση που έφερε επανάσταση στον τομέα του NLP - μια αρχιτεκτονική νευρωνικών δικτύων υπό το όνομα Μετασχηματιστής (Transformer). Η αρχιτεκτονική αυτή - σε αντίθεση με τα RNNs - δεν βασίζεται σε τεχνικές επανάληψης αλλά μόνο σε έναν μηχανισμό “προσοχής” (attention), που επιτρέπει στο μοντέλο να επικεντρώνει κάθε στιγμή στα πιο σχετικά τμήματα των δεδομένων εισόδου για κάθε έξοδο. Επιπλέον, επεξεργάζεται όλα τα δεδομένα εισόδου μαζί με αποτελεσμα να μην έχει τα προβλήματα παραλληλισμού των προηγουμένων μοντέλων, ενώ πλήττεται πολύ λιγότερο από το πρόβλημα του vanishing gradient - με αποτέλεσμα να λειτουργεί πολύ πιο αποδοτικά κατά την εκπαίδευση μεγάλων δικτύων (Vaswani et al, 2017). Μοντέλα που βασίζονται στην αρχιτεκτονική του Μετασχηματιστή, όπως το GTP και το BERT, υπερκέρασαν πλήρως την απόδοση των καλύτερων υλοποιήσεων προηγούμενων αρχιτεκτονικών - σε τέτοιο βαθμό που, τα τελευταία χρόνια, σχεδόν κάθε νέο μοντέλο αιχμής φαίνεται να βασίζεται στην προσέγγιση αυτή.

Παρακάτω αναφέρονται εν συντομία μερικά βασικά χαρακτηριστικά του Transformer, και ακολουθεί σχήμα που απεικονίζει αναλυτικά την αρχιτεκτονική του μοντέλου:

- τα μοντέλα αρχιτεκτονικής Μετασχηματιστή κατά κύριο λόγο προ-εκπαιδεύονται με μη επιβλεπόμενο τρόπο (unsupervised pre-training). Χρησιμοποιούνται δηλαδή πολύ μεγάλα datasets απλού μη-προεπισημασμένου (unlabeled) κειμένου, τα οποία επιτρέπουν στο μοντέλο να υπολογίσει μια αρχική αξιολόγηση των λέξεων, χωρίς ανθρώπινη παρέμβαση. Κατόπιν, σε ένα δεύτερο στάδιο βελτιστοποίησης (“fine-tuning”), οι υπολογισμένες τιμές (η γνώση δηλαδή που αποκτήθηκε) αξιοποιούνται για την προσαρμογή του μοντέλου σε μια συγκεκριμένη επιθυμητή υπο-εργασία (downstream task).
- η αναπαράσταση των λέξεων που “μαθαίνει” αρχικά το μοντέλο είναι ανεξάρτητη του πλαισίου στο οποίο χρησιμοποιούνται. Δεδομένου ότι οι υπολογιστές δεν μπορούν να κατανοήσουν λέξεις αλλά μόνο αριθμούς, η αναπαράσταση αυτή επιτυγχάνεται με την δημιουργία “διανυσμάτων” για κάθε λέξη - πρακτικά, ακολουθίες πραγματικών αριθμών που αποτυπώνουν τον βαθμό συσχέτισης κάθε λέξης με άλλες - ενώ οι λέξεις με παρεμφερή σημασία αναμένεται να αναπαρίστανται και με παρόμοια διανύσματα. Αυτές οι αναπαραστάσεις των λέξεων ονομάζονται “ενσωματώσεις” (word embeddings).
- παρά την ανεξάρτητη-πλαισίου αναπαράσταση των λέξεων που εφαρμόζει η αρχιτεκτονική Transformer, υφίσταται η ανάγκη κατανόησης των συσχετισμών μεταξύ των λέξεων μίας φράσης ή πρότασης καθώς και της γενικότερης δομής της. Για αυτόν τον σκοπό, χρησιμοποιείται ο μηχανισμός “προσοχής” (attention mechanism) που παρουσιάστηκε το 2014 (Cho et al, 2014) και ο οποίος υπολογίζει και ενημερώνει το μοντέλο για τις λέξεις μιας φράσης που έχουν κάποια νοητική συσχέτιση μεταξύ τους.



Εικόνα 2.6 Αρχιτεκτονική transformer (Vaswani et al, 2017)

Συνοψίζοντας, τα μοντέλα Transformers τείνουν να επικρατήσουν έναντι των προηγούμενων αρχιτεκτονικών επειδή παρουσιάζουν μία σειρά πλεονεκτημάτων όπως η δυνατότητα εκπαίδευσης σε μεγαλύτερο όγκο δεδομένων σε λιγότερο χρόνο, η ανάγκη για μικρότερο όγκο επισημασμένων (labeled) δεδομένων, η μεγαλύτερη ακρίβεια, η δυνατότητα κατανόησης συσχετίσεων λέξεων με απόσταση μεταξύ τους και η πολύ αυξημένη προσαρμοστικότητά τους (Srivastava, 2022).

2.6.1. Εξέλιξη των μοντέλων

Το πρώτο μοντέλο που χρησιμοποίησε την αρχιτεκτονική Μετασχηματιστή ήταν το GPT (Generative Pre-trained Transformer), το οποίο παρουσιάστηκε τον Ιούνιο του 2018. Λειτουργούσε με τον βασικό τρόπο που περιγράφηκε νωρίτερα - χρησιμοποιούσε ένα αρχικό στάδιο προ-εκπαίδευσης πάνω σε μη-επεξεργασμένο κείμενο “μαθαίνοντας” να προβλέπει την επόμενη λέξη σε μία φράση και, σε δεύτερη φάση, προσαρμοζόταν σε συγκεκριμένες εργασίες μέσα από επιβλεπόμενες διαδικασίες βελτιστοποίησης (fine-tuning). Η επόμενη έκδοσή του, το GPT-2, διατήρησε τις ίδιες αρχές προ-εκπαίδευσης αλλά χρησιμοποίησε ένα πολύ μεγαλύτερο όγκο αρχικών κειμένων και επεδίωξε να πετύχει τους ίδιους στόχους παραλείποντας την δεύτερη φάση του fine-tuning (Radford et al, 2019). Η προσέγγιση αυτή επέφερε βελτιωμένα αποτελέσματα σε κάποιες γλωσσικές εργασίες, αλλά όχι σε όλες. Κατέδειξε παρόλα αυτά ότι το μέγεθος του συγκεκριμένου μοντέλου δεν ήταν σε καμία περίπτωση το όριο, και υπήρχαν ακόμα πολλά περιθώρια για ακόμα καλύτερη κατανόηση της φυσικής γλώσσας (Shree, 2020). Αυτό αποδείχτηκε και από το μεταγενέστερο GPT-3, το οποίο εκπαιδεύτηκε σε ένα πολύ μεγαλύτερο dataset σε σχέση με το GPT-2, διαθέτει 100 φορές περισσότερες παραμέτρους (175 δισεκατομμύρια) (Brown et al, 2020), ενώ είναι σε θέση να γράψει κείμενα που δύσκολα διαχωρίζονται από ανθρώπινα και να παράξει κώδικα σε διάφορες προγραμματιστικές γλώσσες για έναν σκοπό που θα του ανατεθεί με φυσική διατύπωση.

Τα μοντέλα GPT χρησιμοποιούν μια “μονοκατευθυντική” προσέγγιση - έχουν δηλαδή πρόσβαση μόνο στο κομμάτι της φράσης που προηγείται (αριστερά) της λέξης που κάθε φορά εξετάζουν. Για να αντιπαρέλθουν αυτόν τον περιορισμό, ερευνητές παρουσίασαν λίγους μήνες μετά το GPT, το μοντέλο Αμφίδρομων Αναπαραστάσεων Κωδικοποιητή από Μετασχηματιστές (Bidirectional Encoder Representations from Transformers - BERT) (Devlin et al, 2018). Το BERT λάμβανε υπόψιν του και τις δύο πλευρές κειμένου εκατέρωθεν της κάθε εξεταζόμενης λέξης κατά την αναπαράστασή της, επιτρέποντας την πολύ καλύτερη εξαγωγή νοημάτων. Παρόλα αυτά, ακριβώς λόγω της αμφίδρομης λειτουργίας του και της πρόσβασής του σε ολόκληρη την φράση, δεν μπορούσε να βασίσει την προεκπαίδευσή του

στην εργασία πρόβλεψης επόμενης λέξης όπως τα προηγούμενα μοντέλα τύπου Transformer. Για την επίλυση του προβλήματος αυτού, εισήχθησαν δύο νέες μη-επιβλεπόμενες εργασίες προεκπαίδευσης: το Μοντέλο Καλυμμένης Γλώσσας (Masked Language Model - MLM), που στοχεύει στην πρόβλεψη λέξεων που αντικαθίστανται κατά τυχαίο τρόπο από ένα σύμβολο [MASK], και η Πρόβλεψη Επόμενης Πρότασης (Next Sentence Prediction - NSP).

Λόγω της εξαιρετικής του απόδοσης σε πολλές γλωσσικές εργασίες, το μοντέλο BERT υιοθετήθηκε από τους ερευνητές και προσαρμόστηκε σε πολλές διαφορετικές απαιτήσεις μέσω τροποποιήσεων στην αρχιτεκτονική του. Για παράδειγμα, το επόμενο μοντέλο RoBERTa (Liu et al, 2019) πρότεινε τρεις βελτιώσεις/αλλαγές: την επιμήκυνση της φάσης προ-εκπαίδευσης για ακριβέστερη εκμάθηση, την μεγέθυνση του όγκου των χρησιμοποιούμενων datasets για ευρύτερη κατανόηση, και την αύξηση του μεγέθους του πακέτου δεδομένων που το μοντέλο προσπελαύνει κάθε φορά (batch size) για βελτίωση της απόδοσης τόσο του MLM, όσο και της παράλληλης επεξεργασίας.

Από την άλλη πλευρά, παρά τα αποδεδειγμένα πλεονεκτήματα των μεγαλύτερων μοντέλων, ένα μικρότερο μοντέλο εκπαιδεύεται γρηγορότερα και επίσης παράγει αποτελέσματα γρηγορότερα - ενώ αν είναι αρκετά μικρό, μπορεί να χρησιμοποιηθεί μέχρι και σε φορητές ή εξυπνες συσκευές (IoT). Μία τεχνική προς την κατεύθυνση της συρρίκνωσης των μεγάλων δικτύων είναι η “απόσταξη γνώσης” (knowledge distillation). Συγκεκριμένα, είναι η μέθοδος κατά την οποία ένα μικρό μοντέλο-μαθητής εκπαιδεύεται να αναπαράγει την συμπεριφορά μιας μεγαλύτερης έκδοσης του εαυτού του (δάσκαλος). Παράδειγμα τέτοιου μοντέλου είναι το DistilBERT (Sanh et al, 2019) το οποίο αποτελεί “απεσταγμένη” έκδοση του BERT και επιτυγχάνει 97% της αποτελεσματικότητάς του σε σε συγκεκριμένες μετρήσεις, όντας όμως 40% μικρότερο σε μέγεθος και 60% γρηγορότερο στην εξαγωγή συμπερασμάτων.

Το 2020, προτάθηκε ακόμα μία παραλλαγή του BERT - το μοντέλο “BERT Ενισχυμένης Αποκωδικοποίησης με Απεμπλεγμένη Προσοχή” (Decoding-enhanced BERT with disentangled attention - DeBERTa) (He et al, 2020). Το DeBERTa βελτιώνει τα προηγούμενα μοντέλα με δύο τρόπους. Ο πρώτος είναι η Απεμπλεγμένη Προσοχή, βάσει της οποίας κάθε λέξη αναπαρίσταται με δύο διανύσματα, ξεχωριστά για το περιεχόμενο και για την θέση της εντός του κειμένου, αντί ενός που συνυπολογίζει και τις δύο παραμέτρους. Ο δεύτερος τρόπος είναι ο βελτιωμένος αποκωδικοποιητής, ο οποιος συνυπολογίζει την απόλυτη θέση των κρυφών λέξεων [MASK] εντός του κειμένου, αμέσως πριν επιχειρήσει

την πρόβλεψή τους κατά την προεκπαίδευση. Συγκρινόμενο με το μοντέλο RoBERTa, το DeBERTa αποδίδει καλύτερα, εκπαιδευμένο μόλις στα μισά αρχικά δεδομένα.

Τέλος, αξίζει να αναφερθεί ότι έχουν δημιουργηθεί πολλές εκδόσεις του BERT εξειδικευμένες και προσαρμοσμένες στις ανάγκες συγκεκριμένων γλωσσών. Από την άλλη, το XML (Lample et al, 2019) στοχεύει σε ένα διαγλωσσικό μοντέλο το οποίο θα μπορεί να προεκπαιδεύεται και σε πολυγλωσσικά κείμενα και να επιτυγχάνει βελτιωμένα αποτελέσματα σε εργασίες όπως η μετάφραση κειμένων.

2.6.2. Tokenizers

Ο τομέας της Επεξεργασίας Φυσικής Γλώσσας (NLP) χρησιμοποιεί γλωσσολογικά αλλά και μαθηματικά εργαλεία, προκειμένου να γεφυρώσει το χάσμα μεταξύ της ανθρώπινης γλώσσας και αυτής των υπολογιστών, κατατμίζοντας και μετατρέποντας την φυσική επικοινωνία σε κομμάτια δεδομένων που μπορούν να γίνουν κατανοητά από τα μοντέλα (Burchfiel, 2022). Παρόλο όμως που η ανθρώπινη επικοινωνία διέπεται από καθορισμένα σύνολα γλωσσικών κανόνων, πολλές φορές στην καθημερινότητα αυτοί οι κανόνες κάμπτονται ή αγνοούνται λόγω ιδιαιτεροτήτων όπως οι συντομεύσεις, η αργκό, οι ευφημισμοί, ακόμα και μέσα από τα λάθη. Χρησιμοποιώντας όλο και μεγαλύτερα datasets, οι ερευνητές εκπαιδεύουν τα μοντέλα NLP να “εξοικειώνονται” με αυτές τις ιδιαιτερότητες της φυσικής γλώσσας και να τις διαχειρίζονται με αποτελεσματικότερο τρόπο.

Η Αναγνώριση Λεξικών Μονάδων ή αλλιώς Tokenization, στα πλαίσια του NLP είναι η διαδικασία κατά την οποία πρωτογενή γλωσσικά δεδομένα μεγαλύτερου μήκους όπως προτάσεις, παράγραφοι ή και ολόκληρα κείμενα κατατμίζονται σε μικρότερες γλωσσικές μονάδες (συνήθως λέξεις, αλλά πολλές φορές και σε ακόμα μικρότερα κομμάτια - subwords), στις οποίες είναι ευκολότερο για τους υπολογιστές να αποδώσουν νόημα. Το tokenization είναι μία σημαντική διαδικασία αφού διευκολύνει τα μοντέλα να κατανοήσουν, τόσο το νόημα κάθε λέξης, όσο και τον τρόπο που αλληλοεξαρτώνται οι λέξεις μέσα στο κείμενο, μέσω λειτουργιών όπως ο υπολογισμός της συχνότητας εμφάνισης των λέξεων αλλά και η παρακολούθηση των σημείων όπου εμφανίζονται. Πρακτικά, ο κύριος σκοπός του tokenization είναι να συμβάλει στην δημιουργία ενός λεξικού από τις μοναδικές γλωσσικές μονάδες που συνάντησε στο κείμενο κατά την φάση της εκπαίδευσης, και με τις οποίες το κείμενο να αναπαρασταθεί (Pai, 2020).

Φυσικά, όπως κάθε διαδικασία, έτσι και το tokenization έχει να αντιμετωπίσει προκλήσεις. Μερικές τέτοιες είναι: οι γλώσσες που δεν χρησιμοποιούν αλφάβητα αλλά

απεικονίσεις φθόγγων ή εννοιών (όπως η κινέζικη, η ιαπωνική και η κορεάτικη), τα σύμβολα που αλλάζουν το νόημα των λέξεων (όπως για παράδειγμα τα νομισματικά) και η χρησιμοποίηση αποστρόφων.

Για τις ανάγκες της Επεξεργασίας Φυσικής Γλώσσας, διακρίνουμε τρεις βασικούς τύπους tokenization :

- σε επίπεδο λέξης (word level): πρόκειται για τον πιο δημοφιλή αλγόριθμο tokenization. Χρησιμοποιεί έναν προεπιλεγμένο χαρακτήρα διαχωρισμού (delimiter), συνήθως αυτόν του κενού διαστήματος (whitespace), βάσει του οποίου μετατρέψει το κείμενο σε μια ακολουθία από λέξεις-tokens. Το βασικό μειονέκτημα αυτών των tokenizers είναι ότι αναπόφευκτα κάποιες νέες λέξεις που θα συναντήσουν δεν θα υπάρχουν στο λεξικό που δημιουργήθηκε κατά την εκπαίδευση. Τέτοιες λέξεις αναγκαστικά αντικαθίστανται από ένα γενικό token UNK (εκ του “unknown” - άγνωστο), το οποίο αφενός αφαιρεί από τις λέξεις τις κάθε πληροφορία για το νόημά τους, αφετέρου αποδίδει σε όλες αυτές τις λέξεις την ίδια αναπαράσταση. Τέλος, τα λεξικά που προκύπτουν από τους word tokenizers τείνουν να είναι πολύ μεγάλα, ανάλογα του πολύ μεγάλου μεγέθους κειμένων πάνω στα οποία εκπαιδεύονται τα σύγχρονα μοντέλα.
- σε επίπεδο χαρακτήρα (character level): οι character tokenizers προσπαθούν να επιλύσουν τα προβλήματα αυτών που λειτουργούν σε επίπεδο λέξης, επιλέγοντας μια διαφορετική προσέγγιση. Χρησιμοποιούν ένα πολύ μικρό λεξικό, μεγέθους ίσου με το μέγεθος του εκάστοτε αλφαβήτου (για παράδειγμα 24 για τα ελληνικά ή 26 για τα αγγλικά) και αναπαριστούν κάθε λέξη ως μια σειρά από χαρακτήρες, κάτι το οποίο αποκλείει την πιθανότητα να προκύψουν σε άγνωστες λέξεις σε επόμενα στάδια. Το μειονέκτημα αυτών των tokenizers είναι πως, για παράδειγμα μία πρόταση δέκα λέξεων των δέκα χαρακτήρων έκαστη, που πριν θα παρήγαγε 10 αναπαραστάσεις στην έξοδο (όσες οι λέξεις), πλέον θα παράγει 100 (όσοι οι χαρακτήρες) - κάτι που καθιστά πιο πολύπλοκη την διαδικασία εξαγωγής νοήματος από την διασύνδεση και την θέση των tokens.
- σε επίπεδο υπολέξης (subword level ή n-gram): αυτή η μέθοδος tokenization επιχειρεί να πετύχει μία μέση οδό μεταξύ των άλλων δύο προσεγγίσεων. Αντί να χωρίζει το κείμενο σε λέξεις ή χαρακτήρες, το χωρίζει σε υπολέξεις (ή χαρακτήρες n γραμμάτων / n-grams). Για παράδειγμα, μια λέξη όπως η “ομορφότερη” θα μπορούσε να χωριστεί σε δύο tokens “ομορφό” και “τερη”. Η προσέγγιση αυτή χρησιμοποιείται κατά κόρον από τα μοντέλα τύπου Transformer που έχουν επικρατήσει τα τελευταία χρόνια, επειδή επιτυγχάνουν μια ισορροπία μεταξύ του μεγέθους του λεξικού, της πολυπλοκότητας

εξόδου και της δυνατότητας όσο το δυνατόν ακριβέστερης αναπαράστασης των πρωτογενών κειμένων εισόδου. Ο δημοφιλέστερος αλγόριθμος για subword tokenization στα μοντέλα τύπου μετασχηματιστή είναι ο Κωδικοποιητής Ζεύγους Χαρακτήρων (Byte Pair Encoding - BPE) (Shibata et al, 1999) , ο οποίος μέσα από διαδοχικές εκτελέσεις και ξεκινώντας από ένα λεξικό μόνο με χαρακτήρες, αναγνωρίζει το ζεύγος χαρακτήρων με την μεγαλύτερη συχνότητα στο κείμενο, το προσθέτει στο λεξικό του ως νέο subword και επαναλαμβάνει την διαδικασία "έλεγχος-προσθήκη" μέχρι το λεξικό να αυξηθεί στο επιθυμητό μέγεθος.

2.7. Τεχνικές

Παρακάτω παρουσιάζεται μια σύντομη περιγραφή των δύο βασικών τεχνικών που χρησιμοποιούνται σε αυτήν την εργασία για την ανάλυση συναισθήματος - τα Συμπεράσματα Φυσικής Γλώσσας (Natural Language Inference) και η Ταξινόμηση Μηδενικής Βολής (Zero-shot Classification).

2.7.1. Natural Language Inference - NLI

Η αναγνώριση της συμφωνίας και της αντίθεσης είναι θεμελιώδης για την σωστή επεξεργασία της φυσικής γλώσσας, και απαραίτητη για πλήθος διεργασιών όπως η αναζήτηση πληροφοριών, η σημασιολογική ανάλυση και και η αιτιολόγηση. Natural language inference ονομάζεται η διαδικασία κατά την οποία προσδιορίζεται αν μια υπόθεση (hypothesis) είναι σωστή/σύμφωνη (entailment), λανθασμένη/αντίθετη (contradiction) ή ουδέτερη (neutral) αναφορικά με μια δεδομένη δήλωση-προϋπόθεση (premise). Μερικά παραδείγματα:

Πίνακας 2.1 Παραδείγματα NLI (Conneau, 2018)

Premise	Label	Hypothesis
"Υπάρχουν πολλές καλύτερες μηχανές στην αγορά αυτή τη στιγμή ."	Contradiction	*"Αυτή είναι η πιο γρήγορη μηχανή , δεν θα βρεις καλύτερη μηχανή ."*
"Και χάρηκα που τα είπαμε ."	Neutral	*"Σου μιλάω κάθε μέρα ."*
"Η ταχυδρομική υπηρεσία ήθελε να μειώσει την συχνότητας παράδοσης ."	Entailment	*"Η ταχυδρομική υπηρεσία θα μπορούσε να είναι λιγότερο συχνή ."*

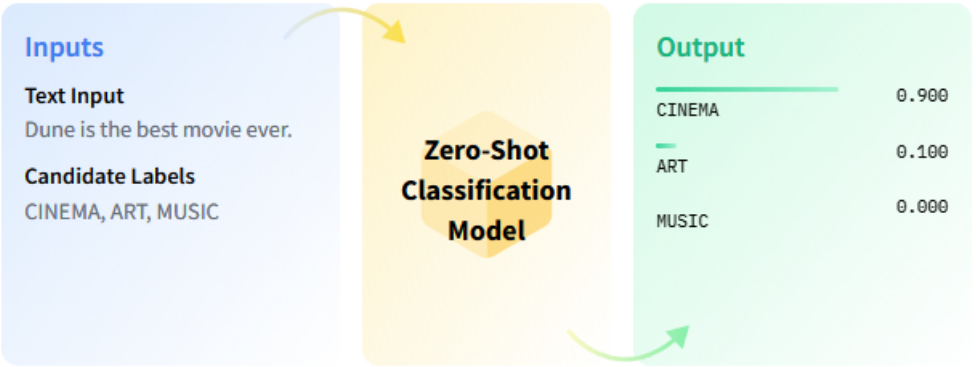
Μερικά πολύ διαδεδομένα datasets με προαξιολογημένα (labeled) ζεύγη προτάσεων για χρήση από μοντέλα συμπερασμάτων είναι:

- το SNLI (Bowman, 2015) από ερευνητές του Πανεπιστημίου του Stanford, με 570 χιλιάδες ζεύγη προτάσεων στην αγγλική γλώσσα. Σκοπός του ήταν να προσφέρει ένα πολύ μεγαλύτερο dataset σε σχέση με τα ως τότε υπάρχοντα (που περιείχαν το πολύ μερικές χιλιάδες παραδειγμάτων), το οποίο θα αποτελείται από φράσεις γραμμένες από ανθρώπους με φυσικό τρόπο και αξιολογημένες επίσης από ανθρώπους, και άρα θα ανταποκρίνεται πολύ καλύτερα στις ανάγκες εκπαίδευσης των σύγχρονων μοντέλων φυσικής γλώσσας.
- το MultiNLI (Williams, 2017) με 430 χιλιάδες ζεύγη, από δέκα διαφορετικά είδη γραπτού και προφορικού λόγου - σε αντίθεση με το SNLI, όπου όλες οι προτάσεις συγγράφηκαν ως μέρος μιας εργασίας δημιουργίας περιγραφών για εικόνες, και άρα δεν προσφέρονταν για αποτύπωση μη-οπτικών εννοιών όπως ο χρόνος ή οι πεποιθήσεις. Σκοπός του MultiNLI ήταν η πληρέστερη αποτύπωση της πολυπλοκότητας της αγγλικής γλώσσας και η δημιουργία ενός πιο απαιτητικού (σε σχέση με το SNLI) σημείου αναφοράς και αξιολόγησης των δυνατοτήτων των μελλοντικών γλωσσικών μοντέλων.
- το XNLI (Conneau, 2018), το οποίο δημιουργήθηκε για να διευρύνει τις δυνατότητες κατανόησης των πολυγλωσσικών μοντέλων, και περιέχει ένα σύνολο 7500 ζευγών προτάσεων από τις ίδιες πηγές με το MultiNLI dataset και αξιολογημένων από τους ίδιους ανθρώπους, και ακολούθως μεταφρασμένο από τα αγγλικά σε 14 επιπλέον γλώσσες (σύνολο 112.500 ζεύγη), μεταξύ των οποίων και τα ελληνικά. Η ύπαρξη της υπόθεσης σε 15 γλώσσες για κάθε μεμονωμένη δήλωση (premise), είναι αυτή που επιτρέπει την αναγνώριση της συμφωνίας ή αντίθεσης μεταξύ προτάσεων διαφορετικών γλωσσών και ταυτόχρονα δημιουργεί ένα δυνητικό αριθμό άνω του 1,5 εκατομμυρίων ζευγών. Επιπλέον, οι μεταφράσεις πραγματοποιήθηκαν από επαγγελματίες, και όχι με κάποιον αυτόματο τρόπο, συντελώντας στην φυσικότητα των προτάσεων. Σημειώνεται ότι τα παραδείγματα του προηγούμενου πίνακα προέρχονται από το ελληνικό τμήμα του XNLI, ενώ και το μοντέλο που χρησιμοποιείται στην παρούσα διπλωματική (DeBERTa) έχει βελτιστοποιηθεί (fine-tuning) στο συγκεκριμένο dataset.

2.7.2. Zero-shot classification

Ταξινόμηση μηδενικής βολής (zero-shot) ονομάζεται η διεργασία κατά την οποία ένα μοντέλο καλείται να εκτελέσει μια ταξινόμηση για την οποία δεν έχει εκπαιδευτεί - δεν έχει τροφοδοτηθεί δηλαδή νωρίτερα με προαξιολογημένα παραδείγματα της ταξινόμησης

που του ζητείται να πραγματοποιήσει. Παρεμφερείς διεργασίες είναι οι “one-shot” και “few-shot” όπου στο μοντέλο παρέχονται αντιστοίχως ένα ή μερικά ολοκληρωμένα παραδείγματα. Οι διεργασίες αυτές αναδεικνύονται ως χρήσιμα εργαλεία ειδικά στα πιο πρόσφατα μεγάλου μεγέθους μοντέλα - όπου ο τεράστιος όγκος των πρωτογενών κειμένων πάνω στα οποία εκπαιδεύονται, μπορεί να αντισταθμίσει την μερική έλλειψη ή και πλήρη απουσία “ακριβών” (σε δυσκολία και πόρους που απαιτούνται για την συγκέντρωσή τους) προαξιολογημένων (labeled) δεδομένων - και γενικά τείνουν να αποδίδουν καλύτερα κλιμακωτά με την αύξηση του μεγέθους των μοντέλων (Hugging Face, 2022).



Εικόνα 2.7 Τεχνική zero-shot classification (Hugging Face, 2022)

Η παρούσα εργασία αξιοποιεί την τεχνική NLI για να πραγματοποιήσει zero-shot classification μεταξύ υποψηφίων labels που θα αντιστοιχούν στα τέσσερα πιο βασικά συναισθήματα.

3. Υλοποίηση

3.1. Twitter API

Το ακρωνύμιο API προέρχεται από τον όρο Application Programming Interface, ή στα ελληνικά Διεπαφή Προγραμματισμού Εφαρμογών. Στην ουσία αποτελεί ένα σύνολο κανόνων που επιτρέπει σε εφαρμογές να επικοινωνούν μεταξύ τους - ο προγραμματιστής δημιουργεί το API πρόσβασης σε έναν εξυπηρετητή (server) και επιτρέπει στον πελάτη (client) να το χρησιμοποιήσει για να αλληλεπιδράσει με τον server.

Σε αντίθεση με τα περισσότερα άλλα κοινωνικά δίκτυα, το Twitter παρέχει ελεύθερη (μέχρι και τον χρόνο συγγραφής αυτού του κειμένου) πρόσβαση στα δεδομένα του για αναπτυξιακούς, εκπαιδευτικούς και ερευνητικούς σκοπούς. Για τις χρήσεις αυτές, το Twitter διαθέτει μία σειρά

από APIs (με το v2 να είναι το βασικό, και το οποίο θα εννοείται στο εξής όταν γίνεται αναφορά σε API χωρίς κάποια άλλη διευκρίνιση), που αποτελούν και την επίσημη δίοδο πρόσβασης στα δημόσια δεδομένα του. Μέσω αυτών των διεπαφών οι ερευνητές μπορούν να συνδεθούν και, συντάσσοντας κατάλληλα αιτήματα-ερωτήματα (requests), να λάβουν απαντήσεις (responses) με πληροφορίες όπως το περιεχόμενο και τον αναγνωριστικό αριθμό των tweets, την ώρα δημιουργίας τους, στατιστικά στοιχεία αλλά και δημόσιες πληροφορίες των χρηστών, ή να εκτελέσουν λειτουργίες εντός του κοινωνικού δικτύου. Το API δεν παρέχει πρόσβαση σε μη-δημόσιες προσωπικές πληροφορίες τρίτων όπως διαγραμμένα tweets, προσωπικά μηνύματα, ημερομηνίες γενεθλίων ή πολιτικές πεποιθήσεις.

Το Twitter API, σύμφωνα με την ίδια την εταιρεία, είναι "ένα σύνολο *προγραμματιστικών σημείων πρόσβασης (endpoints)* που μπορούν να χρησιμοποιηθούν για την κατανόηση ή την δόμηση του διαλόγου στο Twitter. Το API (μέσω αυτών των endpoints) επιτρέπει την αναζήτηση και ανάκτηση, δημιουργία και αλληλεπίδραση με ένα πλήθος διαφορετικών πόρων όπως μηνύματα (tweets), χρήστες, προσωπικά μηνύματα, λίστες, πολυμέσα, spaces (δωμάτια φωνητικών συζητήσεων σε πραγματικό χρόνο), places (συγκεκριμένες ονοματοδοτημένες τοποθεσίες) και δημοφιλή θέματα (trends)".

Όπως προαναφέρθηκε, η κύρια τρέχουσα έκδοση του API είναι η v2, που παρουσιάστηκε τον Αύγουστο του 2020. Για λόγους συμβατότητας, εξακολουθεί να παρέχεται περιορισμένη πρόσβαση στην προηγούμενη έκδοση (v1.1), ενώ διατίθεται ξεχωριστή διεπαφή για διαφημιστικούς σκοπούς (Twitter Ads API). Σύμφωνα με την εταιρεία, η έκδοση v2 "είναι χτισμένη πάνω σε μια σύγχρονη υποδομή, παρέχει νέα και προχωρημένα χαρακτηριστικά και metrics, γρήγορη διαδικασία πρόσβασης στην εισαγωγική βαθμίδα (Essentials) καθώς και μία νέα ειδική βαθμίδα πρόσβασης για ερευνητικούς σκοπούς (Academic)".

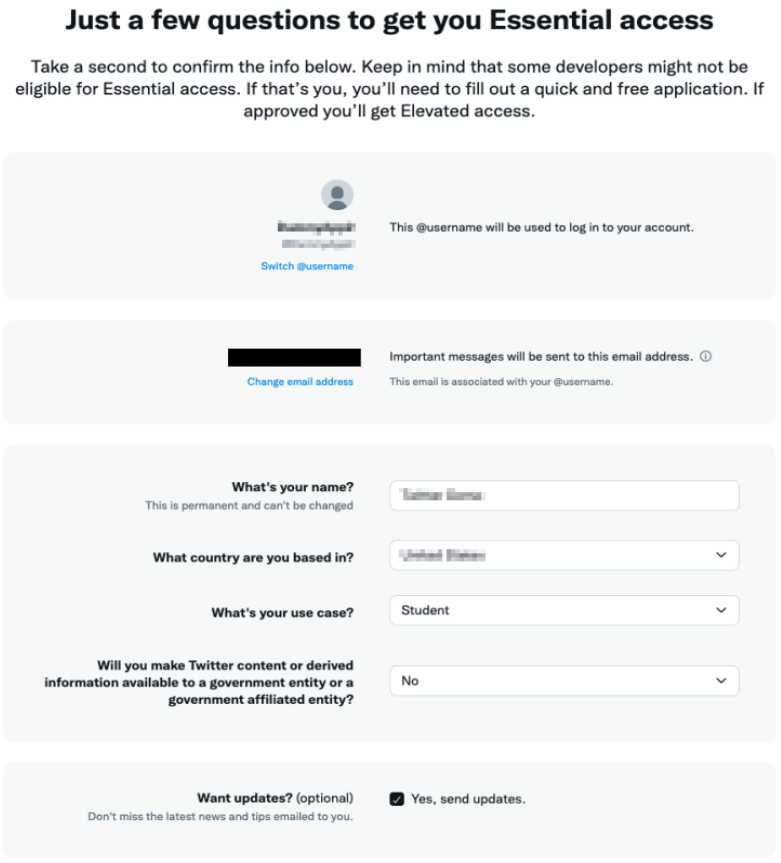
Η πρόσβαση στα προγραμματιστικά εργαλεία του Twitter απαιτεί αρχικά ένα αίτημα για την δημιουργία λογαριασμού προγραμματιστή (developer account), μέσω διαδικασίας που θα επιδειχθεί παρακάτω. Η ολοκλήρωση της διαδικασίας του αιτήματος δίνει στον χρήστη άμεση πρόσβαση στην Πλατφόρμα Προγραμματιστή (Twitter Developer Platform), καθώς και δικαιώματα βασικού επιπέδου (Essentials). Στην τελευταία έκδοση του API (v2), ο χρήστης-προγραμματιστής υποχρεούται να οργανώνει την εργασία του σε Έργα (Projects) και εφαρμογές (Apps) εντός των έργων. Μέσα από την Πλατφόρμα, ο χρήστης μπορεί να βλέπει τα έργα και τις εφαρμογές του, να δημιουργεί νέα και να σβήνει παλαία που τυχόν δεν χρειάζεται, να αλλάζει παραμέτρους τους (όπως ονόματα, περιγραφές, δικαιώματα και άλλα) και να δημιουργεί ή να ανανεώνει διαπιστευτήρια πρόσβασης.

Αναλόγως της επιθυμητής χρήσης, το Twitter διακρίνει τέσσερα επίπεδα πρόσβασης στο API του:

- *Βασικό (Essential)*. Είναι ο γρηγορότερος και ο προτεινόμενος τρόπος από το Twitter για να ξεκινήσει κάποιος την πρόσβασή του στο API v2 και, όπως ήδη αναφέρθηκε, παρέχεται άμεσα χωρίς κάποια εξατομικευμένη εξέταση του αιτήματος. Το επίπεδο αυτό επιτρέπει την ανάκτηση έως 500 χιλιάδων tweets ανά μήνα (με περιορισμό 50 ανά δευτερόλεπτο), ενώ τα αιτήματα αναζήτησης πρέπει να περιορίζονται χρονικά στις τελευταίες 7 ημέρες. Τέλος, υπάρχει όριο ενός Project και ενός App για αυτό το επίπεδο.
- *Ενισχυμένο (Elevated)*. Το επίπεδο αυτό απευθύνεται σε όσους χρειάζονται πάνω από 500.000 tweets το μήνα. Απαιτεί εξατομικευμένη εξέταση του αιτήματος, με αποτέλεσμα να χρειάζεται περισσότερος χρόνος για την απόκτηση της πρόσβασης, αλλά επιτρέπει την ανάκτηση έως 2 εκατομμυρίων tweets ανά μήνα, και την δημιουργία 3 εφαρμογών στα πλαίσια ενός project. Το όριο όγκου των 50 tweets ανά δευτερόλεπτο και ο χρονικός περιορισμός της αναζήτησης εντός των τελευταίων 7 ημερών που ισχύουν στο Βασικό επίπεδο, διατηρούνται και στο Ενισχυμένο.
- *Ακαδημαϊκό-Ερευνητικό (Academic Research)*. Το επίπεδο πρόσβασης αυτό απευθύνεται σε άτομα που συμμετέχουν σε μεταπτυχιακά, διδακτορικά και ερευνητικά προγράμματα, και μπορούν να παρουσιάσουν μια βάσιμη αιτιολόγηση της ανάγκης τους για πρόσβαση σε πολύ μεγαλύτερο όγκο δεδομένων του Twitter. Όπως και για το Ενισχυμένο επίπεδο, απαιτείται εξατομικευμένη εξέταση του αιτήματος αλλά και η παροχή επιπλέον πληροφοριών και τεκμηρίωσης. Σε περίπτωση έγκρισης του αιτήματος, το επίπεδο αυτό παρέχει την δυνατότητα άντλησης έως 10 εκατομμυρίων tweets ανά μήνα από το API v2, με μέγιστο ρυθμό 250 μηνυμάτων ανά δευτερόλεπτο. Επιπρόσθετα, χορηγείται η δυνατότητα αναζήτησης στο πλήρες αρχείο δημοσίων μηνυμάτων του Twitter από το 2006 και μετά, χωρίς δηλαδή τον χρονικό περιορισμό των 7 τελευταίων ημερών. Τέλος, ο εγκεκριμένος χρήστης μπορεί να αξιοποιήσει περισσότερες και πιο λεπτομερείς παραμέτρους αναζήτησης και φιλτραρίσματος των δεδομένων.
- *Εταιρικό (Enterprise)*. Το επίπεδο αυτό επιτρέπει πρόσβαση σε περισσότερα από 10 εκατομμύρια tweets ανά μήνα και απευθύνεται σε εταιρείες και οργανισμούς, παρέχοντας την ευρύτερη δυνατή πρόσβαση και αποκλειστικό προσωπικό για την εταιρική διαχείριση και την τεχνική υποστήριξη των συγκεκριμένων πελατών.

Μεταξύ των επιπέδων αυτών υπάρχουν πολλές ακόμα διαφορές σε παραμέτρους όπως η πρόσβαση σε παλαιότερες και διαφημιστικές διεπαφές, ο αριθμός και το μήκος των κανόνων που μπορούν να οριστούν κατά την πρόσβαση πραγματικού χρόνου στα δεδομένα (μέσω του Filtered stream), το μέγιστο μήκος ερωτημάτων και άλλες. Επίσης, το Twitter έχει ξεχωριστά και πολύ αναλυτικά όρια για κάθε έκδοση του API του (1.1, 2.0, Ads) και για κάθε επιτελούμενη λειτουργία (για παράδειγμα δημιουργία tweets, διαγραφή, αναζήτηση, ορισμός σελιδοδεικτών, καταμέτρηση likes ή άλλων metrics κτλ.) ως προς τον αριθμό και την συχνότητα των αιτημάτων που επιτρέπει να δεχτεί, για να διασφαλίζεται η απρόσκοπτη παροχή των υπηρεσιών του. Τα όρια αυτά σε επίπεδο εφαρμογής αλλά και χρήστη (διαθέσιμα στην διεύθυνση https://developer.twitter.com/en/docs/twitter-api/rate-limits) και οι προγραμματιστές μπορούν να έχουν την εποπτεία των ορίων που αφορούν τις εφαρμογές τους μέσω του πίνακα ελέγχου της Πλατφόρμας Προγραμματιστή (Developer Dashboard).

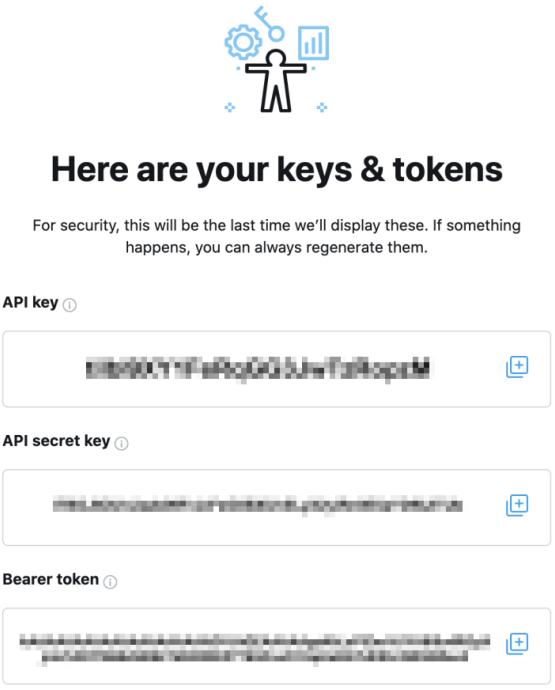
Στην επόμενη εικόνα φαίνονται οι συντομες ερωτήσεις που καλείται να απαντήσει ο ενδιαφερόμενος για πρόσβαση Βασικού επιπέδου. Ακολουθεί η αποδοχή των Όρων της Σύμβασης Προγραμματιστή (Developer Agreement & Policy) και η επιβεβαίωση του λογαριασμού email του ενδιαφερομένου.



Εικόνα 3.1Ερωτηματολόγιο Βασικής πρόσβασης

Αφού αποκτηθεί η πρόσβαση στην πλατφόρμα, και δημιουργηθεί το πρώτο Έργο, απαιτείται η έκδοση και χρήση διάφορων διαπιστευτηρίων για τον έλεγχο της ταυτότητας και του επιπέδου πρόσβασης, ανάλογα με τον τύπο της εφαρμογής. Τα διαπιστευτήρια αυτά παρουσιάζονται παρακάτω, και πρέπει να συμπεριλαμβάνονται σε κάθε αίτημα (request) που αποστέλλεται προς το API:

- *API key & Secret*. Αποτελούν ουσιαστικά το όνομα χρήστη και τον κωδικό για την εφαρμογή (App) μας. Είναι τα πιο βασικά διαπιστευτήρια και απαιτούνται για την έκδοση περαιτέρω τεκμηρίων (tokens) διαπίστευσης.
- *App-only Access Token (γνωστό και ως Bearer token)*. Τεκμήριο διαπίστευσης της εφαρμογής. Χρησιμοποιείται μόνο από εφαρμογές που αποστέλλουν αιτήματα σε endpoints που επιστρέφουν αποκλειστικά δημόσιες πληροφορίες του Twitter (όπως η εφαρμογή της παρούσας διπλωματικής εργασίας). Σε αυτές τις περιπτώσεις, η παροχή του Bearer Token είναι αρκετή και δεν απαιτείται να περιλαμβάνεται στο request καμία άλλη πληροφορία ταυτοποίησης σε επίπεδο χρήστη.



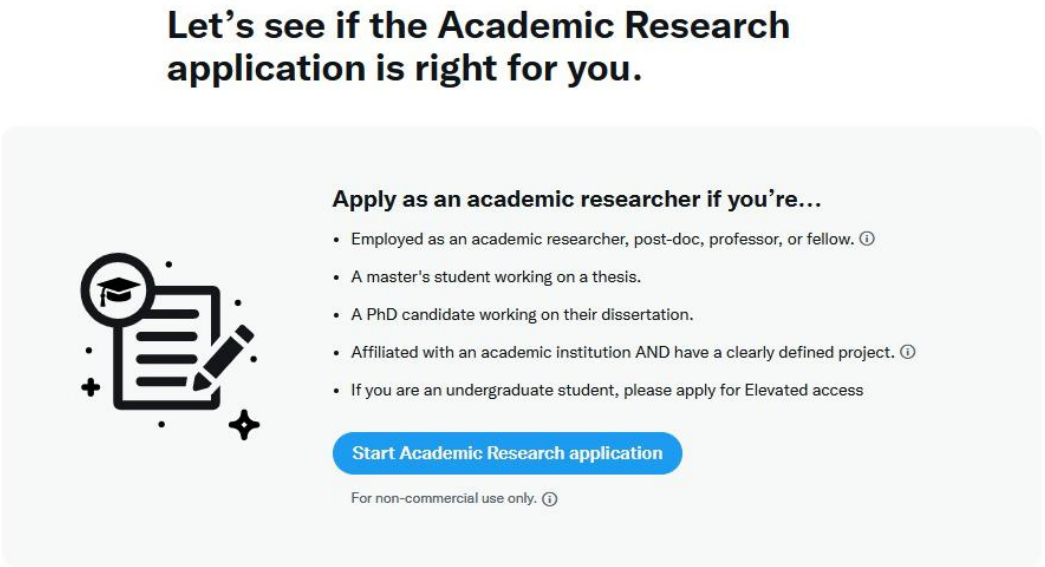
Εικόνα 3.2 Έκδοση διαπιστευτηρίων

- *Access Token & Secret*. Τεκμήριο και κωδικός διαπίστευσης χρήστη. Χρησιμοποιούνται για την ταυτοποίηση του χρήστη της εφαρμογής (App), όταν το endpoint που λαμβάνει το αίτημα επιστρέφει δεδομένα που απαιτούν πιστοποίηση σε επίπεδο χρήστη (OAuth 1.0a User Context). Τα συγκεκριμένα διαπιστευτήρια εκδίδονται πάντα στο όνομα του ιδιοκτήτη της εφαρμογής. Παραδείγματα τέτοιων εφαρμογών είναι εκείνες που

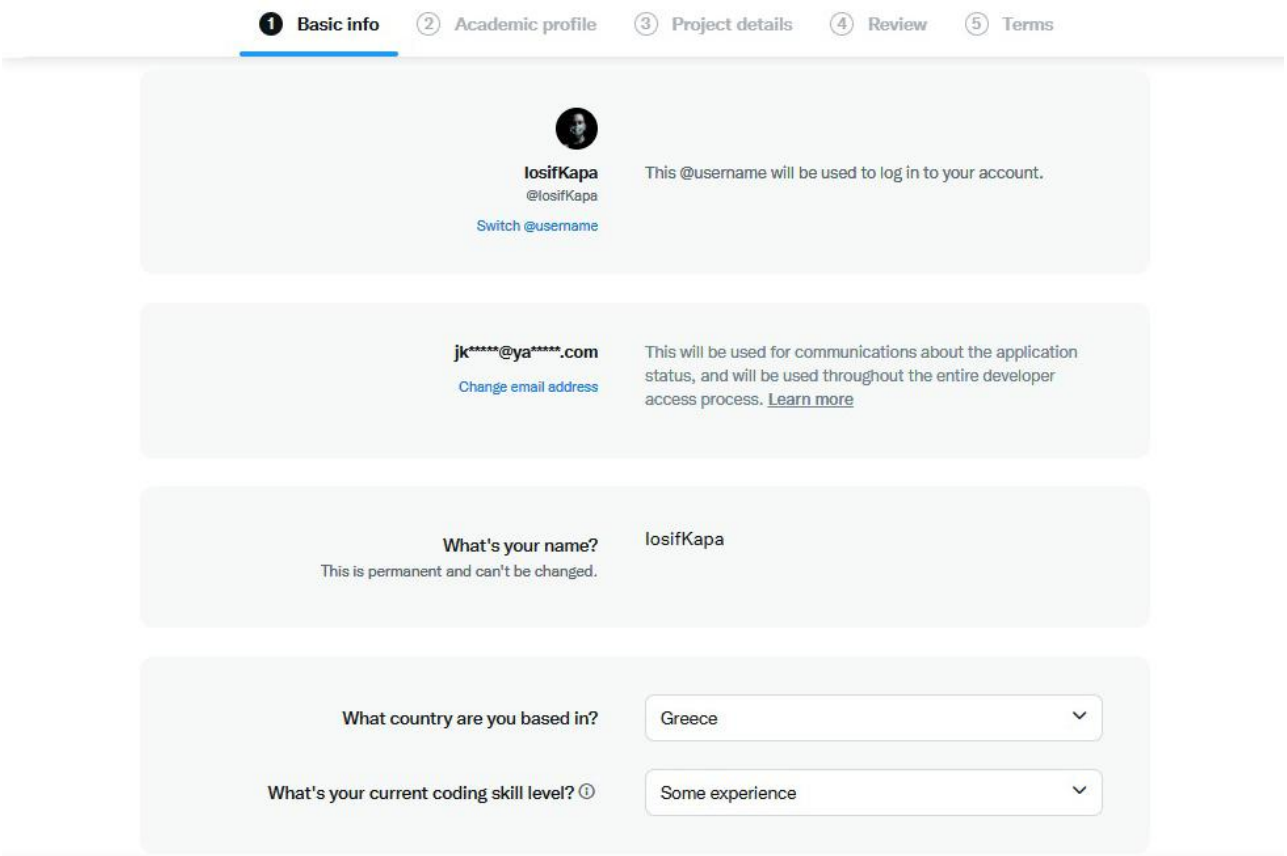
χρησιμεύουν στην διαχείριση προσωπικών λογαριασμών, και άρα έχουν πρόσβαση σε μη-δημόσιες πληροφορίες και δικαιώματα ενεργειών εκ μέρους των λογαριασμών αυτών.

- *Client ID & Client Secret*. Χρησιμεύουν για την έκδοση τεκμηρίων διαπίστευσης χρήστη, έκδοσης OAuth 2.0. Όπως και στην πιστοποίηση έκδοσης 1.0, τα τεκμήρια αυτά αφορούν περιπτώσεις όπου ζητείται πρόσβαση σε προσωπικά δεδομένα ή διαχείριση λογαριασμών τρίτων, αλλά με πιο λεπτομερή παραμετροποίηση και έλεγχο των δικαιωμάτων αυτών. Επιπλέον, η έκδοση 2.0 απλουστεύει κάποιες διαδικασίες αδειοδότησης από πλευράς των τρίτων προς την εφαρμογή.

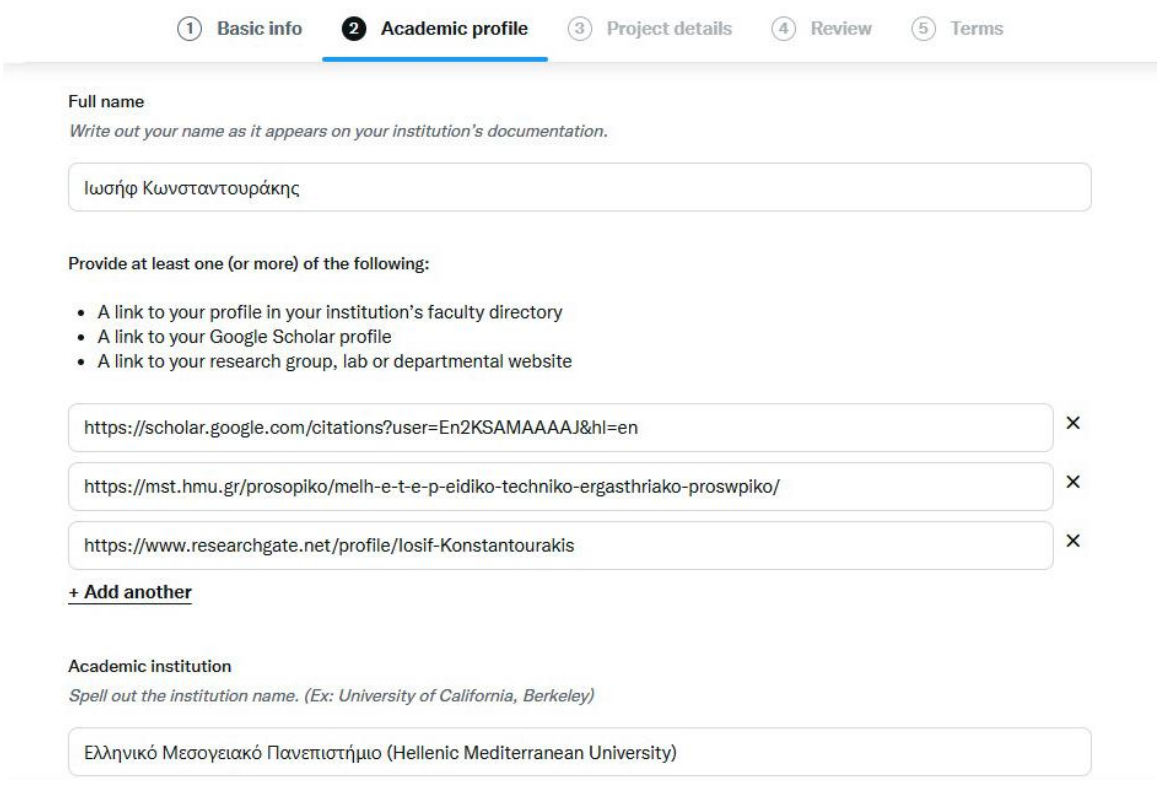
Στις επόμενες εικόνες καταγράφονται τα απαιτούμενα βήματα για την ολοκλήρωση της αίτησης για παροχή πρόσβασης Ακαδημαϊκού-Ερευνητικού επιπέδου στο API του Twitter.



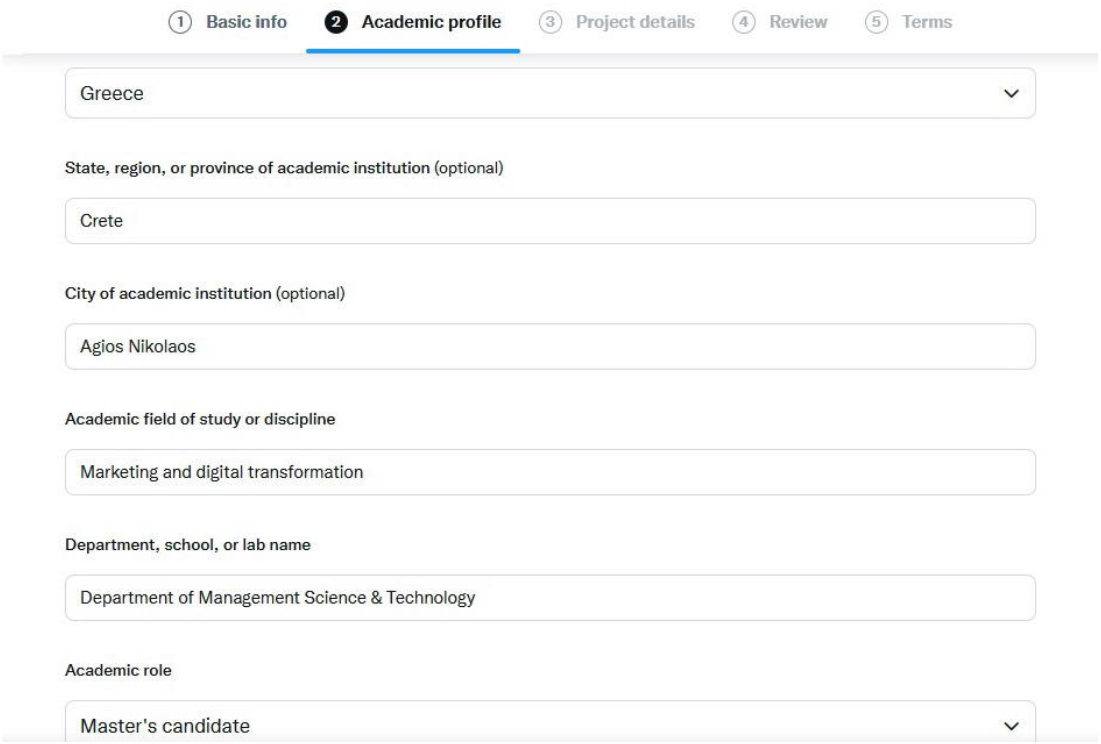
Εικόνα 3.3 Προϋποθέσεις Ακαδημαϊκής πρόσβασης



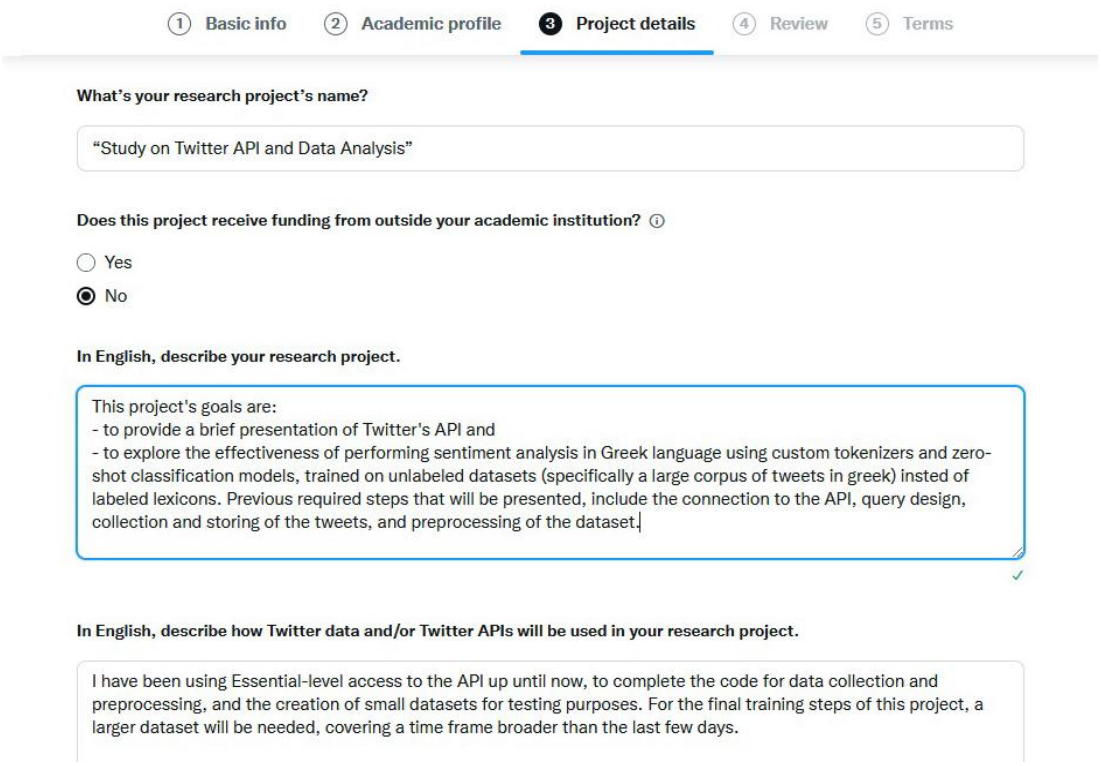
Εικόνα 3.4 Αίτηση Ακαδημαϊκής πρόσβασης - Βασικές πληροφορίες



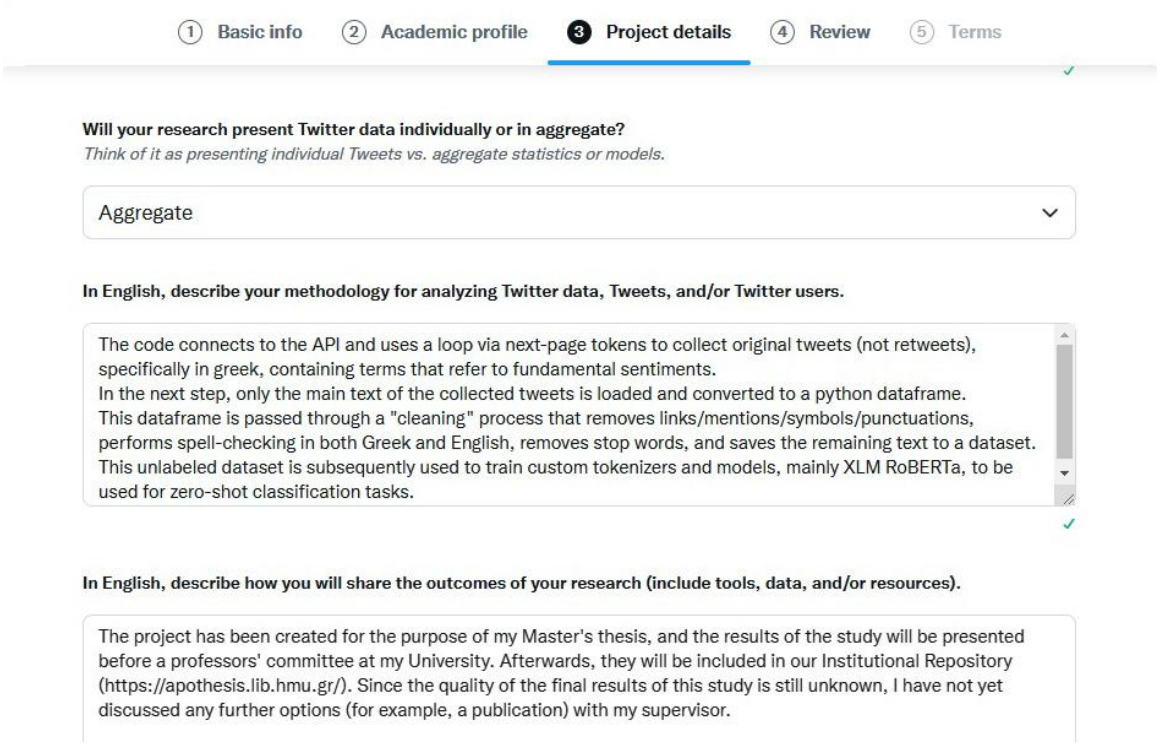
Εικόνα 3.5 Αίτηση Ακαδημαϊκής πρόσβασης - Ακαδημαϊκό προφίλ (1 από 2)



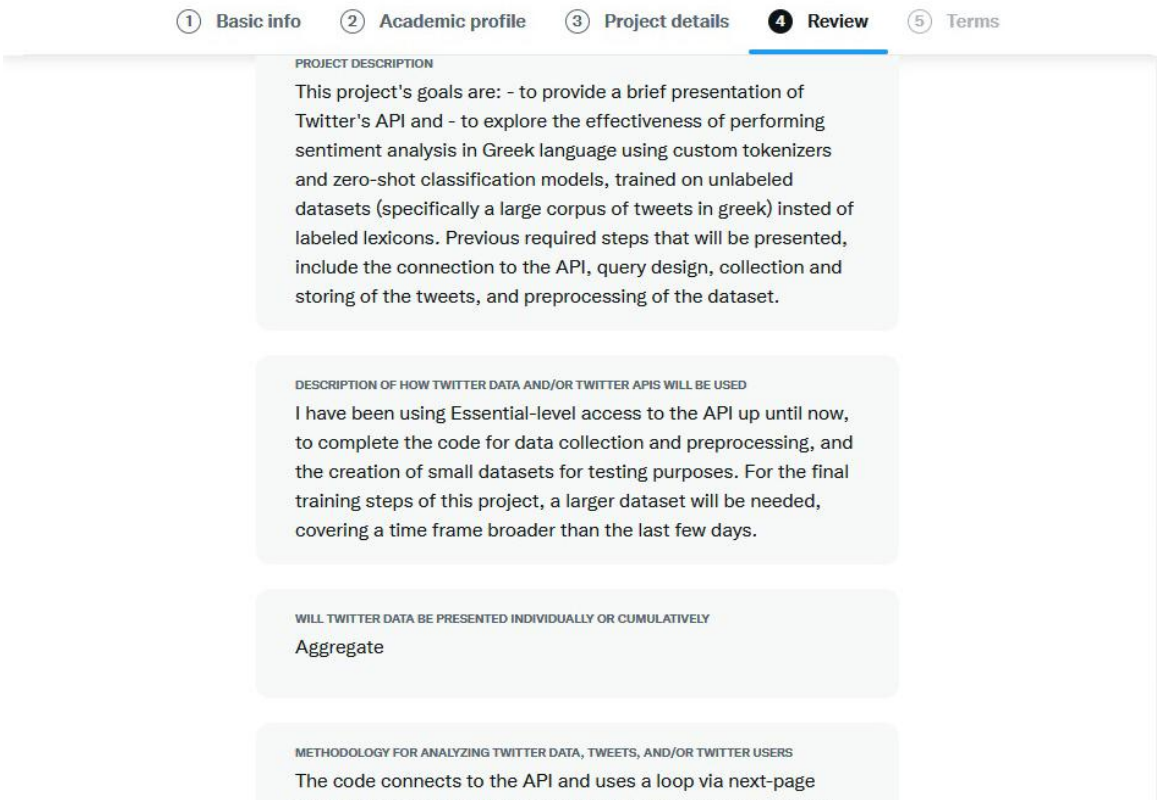
Εικόνα 3.6 Αίτηση Ακαδημαϊκής πρόσβασης - Ακαδημαϊκό προφίλ (2 από 2)



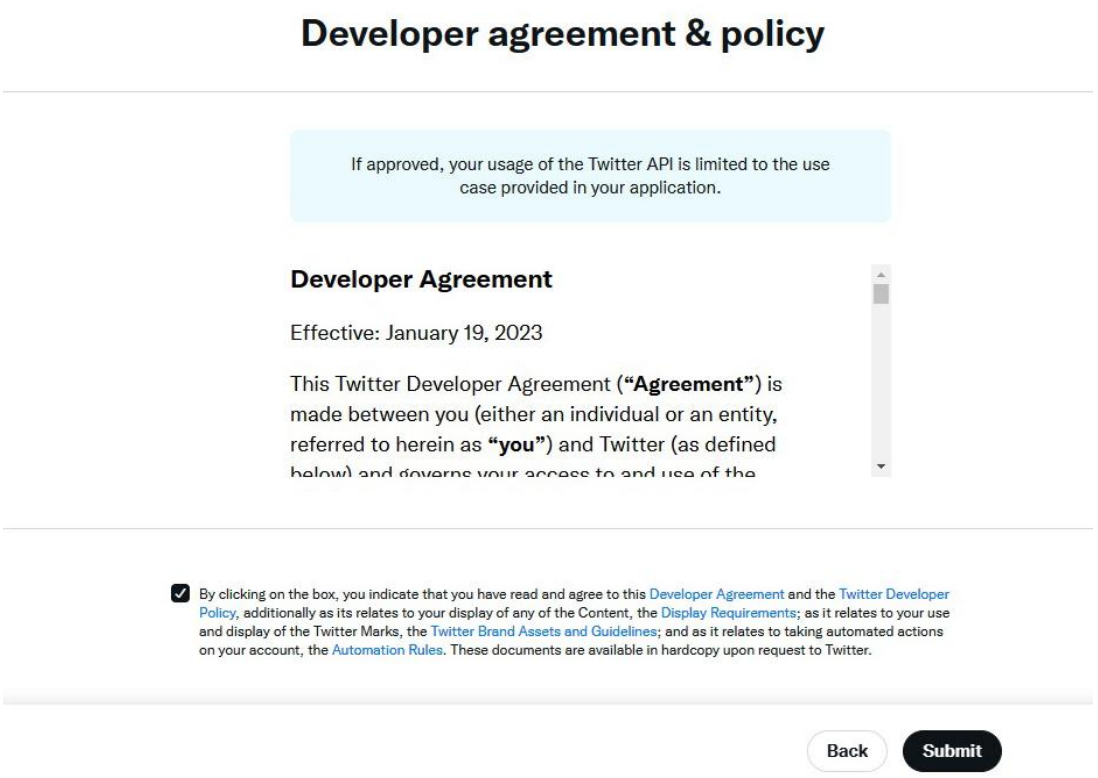
Εικόνα 3.7 Αίτηση Ακαδημαϊκής πρόσβασης - Λεπτομέρειες έργου (1 από 2)



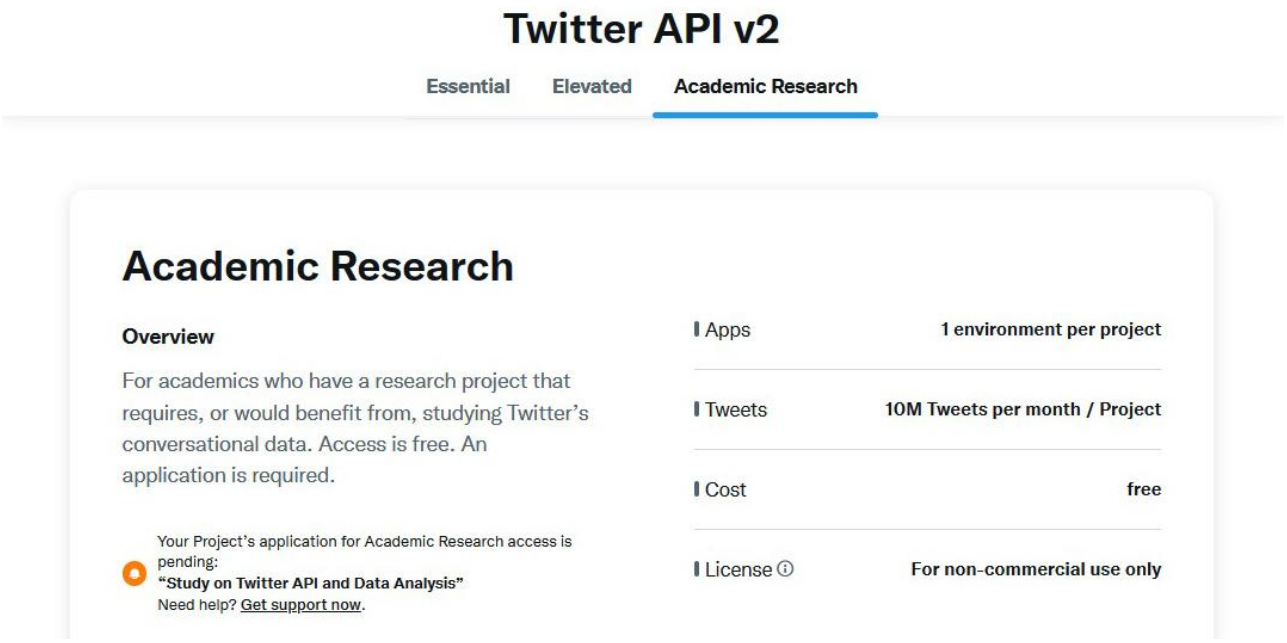
Εικόνα 3.8 Αίτηση Ακαδημαϊκής πρόσβασης - Λεπτομέρειες έργου (2 από 2)



Εικόνα 3.9 Αίτηση Ακαδημαϊκής πρόσβασης - Ανασκόπηση πληροφοριών



Εικόνα 3.10 Αίτηση Ακαδημαϊκής πρόσβασης - Αποδοχή όρων και υποβολή



Εικόνα 3.11 Αίτηση Ακαδημαϊκής πρόσβασης - Ολοκλήρωση

Το Twitter έχει κάνει διαθέσιμο ένα πολύ μεγάλο όγκο πληροφοριών και οδηγιών γύρω από τις προγραμματιστικές διεπαφές του, μέσω σχετικής διαδικτυακής πύλης (https://developer.twitter.com/en/docs/platform-overview). Παρακάτω, θα κάνουμε αναφορά σε συγκεκριμένα

χαρακτηριστικά που αξιοποιήθηκαν σε προγραμματιστικό επίπεδο για τις ανάγκες της παρούσας διπλωματικής εργασίας.

Όπως έχει ήδη αναφερθεί, το API διαθέτει μια πληθώρα από σημεία πρόσβασης (endpoints) που επιτελούν πολλές διαφορετικές λειτουργίες. Σε ό,τι αφορά τα endpoints που επιτελούν λειτουργίες ανάκτησης δεδομένων (γνωστά ως τύπου GET), υπάρχει πάντα ένας πρωτεύοντας τύπος αντικειμένου (όπως μηνύματα, χρήστες, spaces, λίστες, πολυμέσα) που επιστρέφεται στην απάντηση, αλλά μπορούν να ζητηθούν και άλλοι τύποι ως επιπρόσθετοι. Για παράδειγμα, το endpoint πρόσφατης αναζήτησης (“recent search”) επιστρέφει πρωτίστως tweets, όπως και το endpoint φιλτραρισμένης ροής πραγματικού χρόνου (“filtered stream”). Μερικά επιπρόσθετα αντικείμενα που σχετίζονται με το συγκεκριμένο πρωτεύον, θα μπορούσαν να είναι τα πολυμέσα, οι τοποθεσίες ή οι χρήστες, και η περίληψή τους στα αιτήματα-ερωτήματα (requests) γίνεται μέσω της παραμέτρου *expansions* (επεκτάσεις). Κάθε αντικείμενο, πρωτεύον ή επιπρόσθετο, διαθέτει ένα σύνολο από *πεδία* (fields), δηλαδή ιδιότητες, τις οποίες μπορεί να ζητήσει ο χρήστης από το API. Τέτοιες ιδιότητες για το αντικείμενο *tweets* είναι για παράδειγμα ο χρόνος δημιουργίας (created_at), η γλώσσα του κειμένου του (lang) και τα στατιστικά στοιχεία του (public_metrics). Αν το αίτημα προς το endpoint δεν περιλαμβάνει αναφορά σε συγκεκριμένα fields για το αντικείμενο, θα επιστραφεί μόνο ένας μικρός αριθμός προεπιλεγμένων πεδίων - στην περίπτωση των tweets, ο αναγνωριστικός τους αριθμός (id) και το κυρίως κείμενο του μηνύματος (text).

Στο σημείο αυτό, αξίζει να εστιάσουμε λίγο περισσότερο στην αναζήτηση μηνυμάτων (tweets), η οποία είναι ένα σημαντικό χαρακτηριστικό για την ανάδειξη συζητήσεων γύρω από ένα συγκεκριμένο θέμα, συμβάν ή οντότητα. Παρόλο που η δυνατότητα αναζήτησης υπάρχει ενσωματωμένη σε κάθε σελίδα του Twitter μέσω σχετικού πεδίου διαθέσιμου σε όλους τους χρήστες, τα σχετικά endpoints του API προσφέρουν πολύ μεγαλύτερη ευελιξία και ισχύ για την ανεύρεση των επιθυμητών δεδομένων. Το Twitter προσφέρει δύο τέτοια προγραμματιστικά σημεία πρόσβασης, τα οποία ακολουθούν το πρότυπο REST (Representational State Transfer - μεταβίβαση αντιπροσωπευτικής κατάστασης), ένα συγκεκριμένο σύνολο κανόνων λειτουργίας για APIs που ορίζει, μεταξύ άλλων, τον τρόπο δόμησης των αιτημάτων (requests) και των απαντήσεων (responses) - ως εκ τούτου, τα δύο endpoints έχουν κοινό σχεδιασμό και χαρακτηριστικα. Συγκεκριμένα, και τα δύο κάνουν χρήση ενός μοναδικού ερωτήματος σε μορφή URL (Uniform Resource Locator - ουσιαστικά, μιας διαδικτυακής διεύθυνσης) για κάθε αίτημα ανάκτησης tweets. Το ερώτημα αυτό συντάσσεται κάνοντας χρήση παραμέτρων όπως λέξεις-κλειδιά (keywords) και θεματικές επισημάνσεις (hashtags), οι οποίες μπορούν να

συνδυαστούν με boolean λογική (AND/OR) και προτεραιοποίηση βάσει παρενθέσεων, για την περαιτέρω παραμετροποίηση των αποτελεσμάτων. Τα δύο αυτά endpoints είναι τα εξής:

- *Πρόσφατης αναζήτησης (Recent search)*. Προσφέρει προγραμματιστική πρόσβαση σε μηνύματα που δημιουργήθηκαν στο χρονικό πλαίσιο των τελευταίων 7 ημερών πριν το αίτημα. Παρέχεται σε όλους τους προγραμματιστές που έχουν Βασική πρόσβαση και μια εφαρμογή (App) εντός ενός Project. Το endpoint αυτό επιστρέφει έως 100 tweets (σε αντίστροφη χρονολογική σειρά, ξεκινώντας από το πιο πρόσφατο) ανά αίτημα, και περιλαμβάνει τεκμήρια σελιδοποίησης (pagination tokens) τα οποία μπορούν να αξιοποιηθούν προγραμματιστικά για την ολοκληρωμένη λήψη μίας τυχόν μεγαλύτερης απάντησης.
- *Αναζήτησης πλήρους αρχείου (Full-archive search)*. Εισήχθη στην έκδοση v2 του API και είναι διαθέσιμο μόνο στα πλαίσια της Ακαδημαϊκής και της Εταιρικής πρόσβασης. Επιτρέπει την προγραμματιστική πρόσβαση σε ολόκληρο το αρχείο μηνυμάτων του Twitter από τον Μάρτιο του 2006 και μετά, χωρίς χρονικούς περιορισμούς. Επιστρέφει έως 500 tweets ανά αίτημα, χρησιμοποιώντας όπως και το Recent search τεκμήρια σελιδοποίησης για εκτενέστερα σύνολα αποτελεσμάτων.

Πέραν αυτών των δύο βασικών endpoints, το Twitter έχει κάνει διαθέσιμες δύο επιπλέον ομάδες από endpoints για πιο ειδικές χρήσεις:

- Φι*λτραρισμένης ροής (filtered stream)*. Αυτή η ομάδα από endpoints επιτρέπει το φιλτράρισμα μιας ροής από δημόσια μηνύματα του Twitter σε πραγματικό χρόνο (real time). Συγκεκριμένα, οι λειτουργίες τους περιλαμβάνουν την δημιουργία, διαχείριση και εφαρμογή κανόνων φιλτραρίσματος σε πραγματικό χρόνο, πάνω σε μια σταθερή ροή μηνυμάτων που δημιουργούνται εκείνη την στιγμή στο κοινωνικό δίκτυο, χωρίς την ανάγκη αποσύνδεσης από την ροή.
- *Ροής όγκων (volume stream)*. Περιλαμβάνει 2 endpoints που έχουν πάρει το όνομά τους από τον μεγάλο όγκο μηνυμάτων που έχουν σχεδιαστεί να επιστρέφουν, επιλέγοντας με τυχαίο τρόπο ένα υποσύνολο των δημοσίων tweets που δημιουργούνται σε πραγματικό χρόνο. Συγκεκριμένα, το *1% sampled stream* που είναι διαθέσιμο σε όλα τα επίπεδα πρόσβασης, και το *10% sampled stream* που είναι διαθέσιμο μόνο στο Εταιρικό επίπεδο. Τα δύο αυτά endpoints έχουν δημιουργηθεί για να διευκολύνουν την αναγνώριση και παρακολούθηση (monitoring) τάσεων, αντιδράσεων, διεθνών γεγονότων και της γενικότερης παρακολούθησης του δημοσίου αισθήματος (public sentiment).

Είναι χρήσιμο να σημειωθεί ότι το Twitter έχει επίσης αναλυτικούς κανόνες για τον τρόπο που μετράει τον αριθμό των χαρακτήρων στα μηνύματα, αποδίδοντας διαφορετικά βάρη σε ομάδες χαρακτήρων, εικονίδια, και ιδεογράμματα (γλύφους) ανατολικών γλωσσών. Σε κάθε περίπτωση, το API δέχεται μόνο κείμενα σε κωδικοποίηση UTF-8.

Το Twitter παρέχει δείγματα κώδικα σε διάφορες προγραμματιστικές γλώσσες (όπως JavaScript, Python και R) μέσω της διαδικτυακής συνεργατικής πλατφόρμας ανάπτυξης λογισμικών *GitHub*, για να υποστηρίξει τους developers στα πρώτα τους βήματα στο API για όλες τις λειτουργίες που αναφέρθηκαν παραπάνω (https://github.com/twitterdev/Twitter-API-v2-sample-code). Επιπλέον, σε ειδική σελίδα συγκεντρώνει όλες τις βιβλιοθήκες πρόσβασης και τα εργαλεία που έχουν δημιουργηθεί τόσο από το ίδιο το Twitter όσο και από τρίτους προγραμματιστές για πρόσβαση στα APIs, για να καθίσταται πιο εύκολη η αναζήτησή τους (https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries/v2).

Κλείνοντας, αξίζει να γίνει αναφορά σε δύο ενδιαφέρουσες λειτουργίες που έχει προστεθεί στην έκδοση v2 του API και σε όλα τα endpoints που επιστρέφουν tweets:

- Η λειτουργία σελιδοποίησης που αναφέρθηκε και νωρίτερα, στην περίπτωση που τα αποτελέσματα του αιτήματος που λαμβάνει το API δεν είναι δυνατό να ολοκληρωθούν σε μία και μόνη απάντηση (όπως καθορίζεται από την παράμετρο max_results). Όταν τα αποτελέσματα ξεπερνούν το παραπάνω όριο, στην απάντηση μετά το πρώτο υποσύνολο αποτελεσμάτων περιλαμβάνεται μια επιπλέον παράμετρος "next_token", η οποία μπορεί να χρησιμοποιηθεί ως παράμετρος εισόδου στην επόμενη εκτέλεση του αιτήματός μας στα πλαίσια ενός προγραμματιστικού βρόχου (loop), για να αποσταλεί η επόμενη σελίδα αποτελεσμάτων. Όταν στην απάντηση δεν περιλαμβάνεται η παράμετρος "next_token", μπορεί να εξαχθεί το συμπέρασμα ότι τα αποτελέσματα έχουν εξαντληθεί. Τα αποτελέσματα παρουσιάζονται πάντα με αντίστροφη χρονολογική σειρά, δλδ από το πιο πρόσφατο στο παλαιότερο. Αντιστοίχως υφίσταται και η παράμετρος “previous_token” σε περίπτωση που ο προγραμματιστής επιθυμεί να αναφερθεί στην προηγούμενη σελίδα αποτελεσμάτων.
- Επίσης, το Twitter προσπαθεί να επισημάνει/ομαδοποιήσει κάθε καινούριο tweet σε μία από περίπου 80 (έως τώρα) θεματικές κατηγορίες βάσει του περιεχομένου τους και άλλων χαρακτηριστικών, ως ένα εργαλείο για πιο δημιουργικές αναζητήσεις - αν και σε πολλές περιπτώσεις, λόγω της φύσης των μηνυμάτων, αυτό δεν είναι εφικτό. Μερικές τέτοιες κατηγορίες (annotations) είναι ενδεικτικά: τηλεοπτικές εκπομπές, μέρη, αθλήματα, πολιτικοί, ηθοποιοί, βιβλία, ραδιοφωνικοί σταθμοί, διακοπές/αργίες,

φαγητά, χώρες, ταξίδια, ζώα, έκτακτα συμβάντα και πολλές ακόμα (https://developer.twitter.com/en/docs/twitter-api/annotations/overview).

Για την προγραμματιστική υλοποίηση των επομένων σταδίων της διπλωματικής εργασίας, επιλέχθηκαν τα εξής εργαλεία:

- η Python ως γλώσσα προγραμματισμού. Η Python έχει σχετικά απλό και φιλικό συντακτικό, που την καθιστά εύκολη και στην ανάγνωση και κατανόηση. Επιπλέον διαθέτει πλήθος επιστημονικών βιβλιοθηκών, ιδιαίτερα χρήσιμων για παράδειγμα στην επεξεργασία φυσικής γλώσσας (NLP) και την εξαγωγή στατιστικών δεδομένων. Μετά την R, η Python είναι η δημοφιλέστερη γλώσσα για χρήση στις επιστήμες δεδομένων, αλλά και σε τομείς όπως η τεχνητή νοημοσύνη (artificial intelligence) και η βαθιά μάθηση (deep learning).
- το Jupyter Notebook, ένα δωρεάν λογισμικό που βασίζεται σε ελεύθερα πρότυπα (όπως το JSON), ως περιβάλλον ανάπτυξης. Το Jupyter Notebook διευκολύνει πολύ την ανάπτυξη εφαρμογών σε Python αφού, βασιζόμενο στο γεγονός ότι είναι μια γλώσσα που δεν απαιτεί μεταγλώττιση (compilation) του κώδικα πριν την εκτέλεσή του (interpreted γλώσσες), επιτρέπει στον προγραμματιστή να δομεί και να εκτελεί τον κώδικά του τμηματικά, να εντοπίζει εύκολα προβλήματα ή λάθη, και να βλέπει άμεσα τα επιμέρους αποτελέσματα (κείμενο, γραφήματα) κάτω από κάθε τμήμα κώδικα. Τέλος, επιτρέπει την αποθήκευση των αποτελεσμάτων, μορφοποιημένου κειμένου και πολυμέσων μέσα στο ίδιο αρχείο, διευκολύνοντας τις παρουσιάσεις σε τρίτους.
- η πλατφόρμα Hugging Face για την χρήση μοντέλων μηχανικής μάθησης. Το Hugging Face (*https://huggingface.co/*) παρέχει ελεύθερη πρόσβαση σε ένα ολοκληρωμένο σετ εργαλείων για την ανάπτυξη εφαρμογών σε τομείς που σχετίζονται με την τεχνητή νοημοσύνη, με έμφαση στην επεξεργασία φυσικής γλώσσας NLP, μέσω της βιβλιοθήκης Transformers που έχει δημιουργήσει. Επιπλέον, περιλαμβάνει υποδομές για την φιλοξενία, την (αυτοματοποιημένη μέσω pipelines ή μη) χρησιμοποίηση και τον διαμοιρασμό ολοκληρωμένων προεκπαιδευμένων μοντέλων και datasets. Τέλος, για πολλά από αυτά τα μοντέλα ενσωματώνεται και δυνατότητα επίδειξης της λειτουργικότητάς τους. Κατά την συγγραφή αυτού του κειμένου, το Hugging Face φιλοξενούσε πάνω από από 154.000 μοντέλα και σχεδόν 25.000 datasets.
- το μοντέλο Multilingual mDeBERTa-v3-base-mnli-xnli, το οποίο είναι το μοναδικό πολυγλωσσικό μοντέλο τύπου DeBERTa (το οποίο όπως αναφέρθηκε στην βιβλιογραφική ανασκόπηση, αποτελεί μία από τις πλέον σύγχρονες και αποδοτικές παραλλαγές μοντέλου τύπου BERT) που διατίθεται μέσω της πλατφόρμας Hugging

Face κατά τον χρόνο συγγραφής αυτής της εργασίας και έχει εκπαιδευτεί για εργασίες natural language inference πάνω και σε ελληνικά κείμενα. (Laurer et al, 2022).

Στο Παράρτημα Α παρουσιάζεται ολοκληρωμένος ο κώδικας για τα τέσσερα στάδια της προγραμματιστικής υλοποίησης, μορφοποιημένος και με πλήρη σχολιασμό, συμπεριλαμβανομένων (με αχνό χρώμα γραμματοσειράς) μη ενεργών τμημάτων που μπορούν να χρησιμοποιηθούν για πειραματισμό, εναλλακτικές προσεγγίσεις ή μελλοντική αναφορά. Στις επόμενες ενότητες του κεφαλαίου γίνεται παρουσίαση των σταδίων της υλοποίησης και επεξήγηση των σημαντικότερων τμημάτων του κώδικα.

3.2. Συγκέντρωση tweets

Το πρώτο βήμα στην διαδικασία της ανάλυσης είναι η συγκέντρωση ενός συνόλου πρωτογενών δεδομένων, τα οποία θα χρησιμοποιηθούν τόσο για την παραγωγή ενός λεξικού και τις δοκιμές επέκτασης-βελτίωσης του χρησιμοποιούμενου tokenizer/μοντέλου, όσο και για την ανίχνευση συναισθήματος. Για τους σκοπούς της διπλωματικής εργασίας, τα δεδομένα αυτά θα είναι δημόσια μηνύματα του Twitter. Κατά την επικοινωνία με το API v2 γίνεται χρήση πρόσβασης επιπέδου Essential, αφού μέχρι και την σύνταξη αυτού του κεφαλαίου παρέμενε εκκρεμής από πλευράς Twitter η εξέταση του αιτήματος για Ακαδημαϊκή-Ερευνητική πρόσβαση.

Μετά την εισαγωγή των απαραίτητων βιβλιοθηκών, δημιουργήθηκαν (βάσει οδηγιών και παραδειγμάτων που παρέχει και το ίδιο το Twitter) αντικείμενα και συναρτήσεις που αναλαμβάνουν:

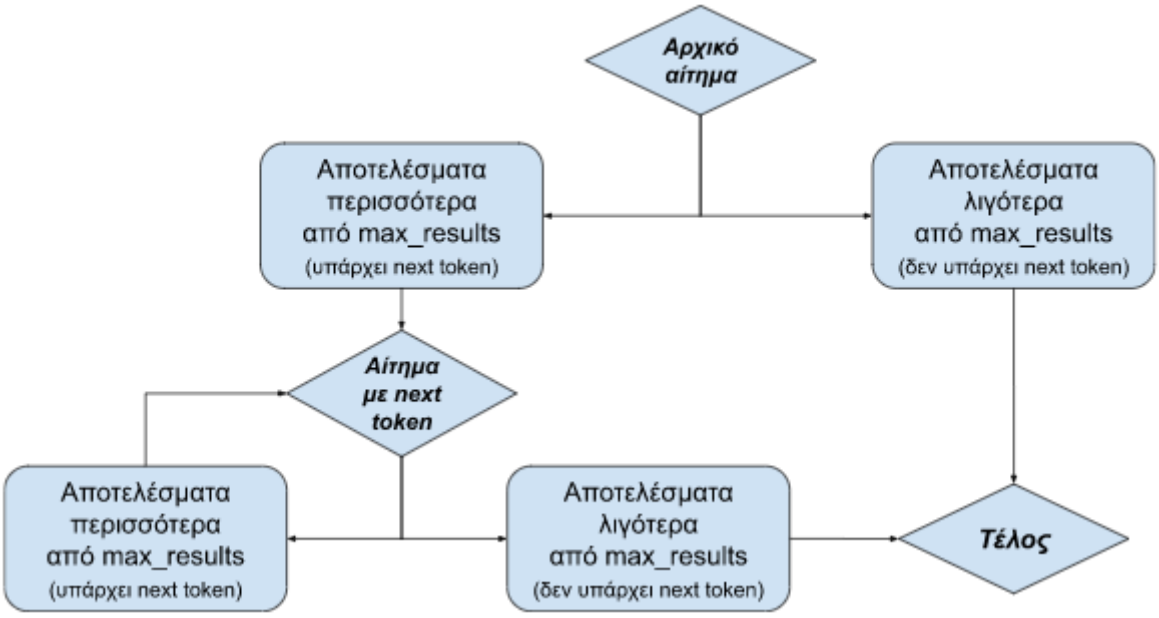
- την αρχικοποίηση του αρχείου αποθήκευσης των επιστρεφόμενων tweets, το οποίο επιλέχθηκε να είναι μορφής .csv (csvFile, csvWriter). Τα αρχεία CSV (comma-separated values) έχουν την απλούστερη δυνατή δομή, χωρίς περιττά στοιχεία, με αποτέλεσμα να είναι ιδανικά για διαχείριση μεγάλου όγκου στοιχείων και για την μεταφορά δεδομένων (εισαγωγή/εξαγωγή) μεταξύ εφαρμογών. Επίσης, η απλή δομή τους τα καθιστά αναγνώσιμα και από ανθρώπους. Ως σύμβολο διαχωρισμού (delimiter) μεταξύ των πεδίων (στηλών) κάθε εγγραφής (γραμμής) του αρχείου επιλέχθηκε η κάθετος (|) αντί του προεπιλεγμένου ελληνικού ερωτηματικού (;), αφού δεν απαντάται σχεδόν ποτέ στο κείμενο των tweets, οπότε ελαχιστοποιούνται οι πιθανότητες να αλλοιωθεί το περιεχόμενο των κειμένων από την προληπτική απαλοιφή του (για να μην επηρεαστεί η δομή του αρχείου .csv κατά την εγγραφή των δεδομένων). Τέλος, επιλέχθηκε η κωδικοποίηση utf-8 για το αρχείο, αφού με αυτήν την κωδικοποίηση επιστρέφονται και τα δεδομένα από τα endpoints του Twitter.

- την αποθήκευση, ανάκληση και μορφοποίηση των διαπιστευτηρίων της εφαρμογής (TOKEN, auth, create_headers). Όπως εξηγήθηκε νωρίτερα, επειδή η εφαρμογή μας στέλνει αιτήματα σε endpoint που επιστρέφει αποκλειστικά δημόσια δεδομένα, το μόνο απαιτούμενο διαπιστευτήριο είναι το *τεκμήριο πιστοποίησης εφαρμογής*, γνωστό και ως Bearer Token.
- τον τμηματικό σχηματισμό του request προς το API, με συνένωση παραμέτρων όπως οι λέξεις-κλειδιά, τα χρονικά όρια αναζήτησης και τα απαιτούμενα πεδία για κάθε tweet της απάντησης (create_url, keyword, start_time_list, end_time_list, query_params). Μεταξύ των πολλών διαθέσιμων πεδίων που μπορούμε να περιλάβουμε στα αιτήματα για δημόσια tweets, επιλέξαμε για αυτό το στάδιο να περιοριστούμε στο αναγνωριστικό του tweet (id), την ημερομηνία δημιουργίας (created_at), το αναγνωριστικό του συγγραφέα (author_id) και φυσικά το κείμενο του tweet (text). Στα επόμενα στάδια, θα χρειαστούμε τελικά μόνο το κείμενο.
- την σύνδεση στο endpoint και την αποστολή του σχηματισμένου request μαζί με τα απαραίτητα διαπιστευτήρια (connect_to_endpoint).
- την διαδοχική εγγραφή των απαντήσεων στο αρχείο .csv σε κατάλληλες στήλες (append_to_csv). Σημειώνεται ότι το Twitter επιστρέφει τα tweets σε μορφή JSON (JavaScript Object Notation). Το JSON είναι ένα ελεύθερο πρότυπο αποθήκευσης και ανταλλαγής δεδομένων, ανεξάρτητο προγραμματιστικής γλώσσας, που δομείται ως ζεύγη ιδιοτήτων-τιμών οργανωμένα σε λίστες. Το πρότυπο JSON χρησιμοποιείται ευρέως επειδή είναι εύκολη τόσο η προσπέλασή του από εφαρμογές, όσο και η ανάγνωσή του από ανθρώπους. Ένα εκτεταμένο παράδειγμα tweet σε μορφή JSON φαίνεται παρακάτω.

```
{
        "id": 464731336310140929,
        "text": "RT @9GAGTweets: From Hulk... hue hue hue - http:\/\/t.co\/bNRmL7uUVf",
        "user": {
                "id": 534017128,
                "name": "jasmine s.",
                "screen_name": "jasminesaff",
                "followers_count": 311,
                "favourites_count": 0,
                "friends_count": 245,
                "verified": false,
                "statuses_count": 10704,
         },
        "retweeted_status": {
                "id": 464717213304627200,
                "text": "From Hulk... hue hue hue - http:\/\/t.co\/bNRmL7uUVf",
                "user": {
                        "id": 471022109,
                        "name": "9GAG Tweets",
                        "screen_name": "9GAGTweets",
                        "followers_count": 251032,
                        "friends_count": 3,
                        "favourites_count": 0,
                        "statuses_count": 124287,
                },
                "retweet_count": 151,
                ”favorite_count": 46,
                "entities": {
                        "hashtags": [  ],
                        "media": [
                                {
                                "media_url": "http:\/\/pbs.twimg.com\/media\/BnMB19sIIAAJv0O.jpg",
                                }
                        ]
                }
        },
        "retweet_count": 0,
        "favorite_count": 0,
        "entities": {
                "hashtags": []
                "media": [
                        {
                        "media_url": "http:\/\/pbs.twimg.com\/media\/BnMB19sIIAAJv0O.jpg",
                        }
                ]
        }
}
```


Εικόνα 3.12 Παράδειγμα tweet σε JSON format

Τελευταίο τμήμα του κώδικα για την λειτουργία συλλογής δεδομένων, είναι ο βρόχος (loop) που αναλαμβάνει την κλήση των παραπάνω συναρτήσεων και την πραγματοποίηση επαναλαμβανόμενων κύκλων σύνδεσης-αιτήματος-λήψης-αποθήκευσης, αξιοποιώντας τον μηχανισμό σελιδοποίησης που διαθέτουν τα endpoints επιστροφής δεδομένων του Twitter, έως ότου εξαντληθούν οι απαντήσεις που αντιστοιχούν στο αίτημά μας. Ταυτόχρονα, μέσω εξωτερικού μετρητή, καταγράφει τον συνολικό αριθμό tweets που λήφθηκαν και αποθηκεύτηκαν και τον εμφανίζει μετά την ολοκλήρωση της διαδικασίας.



Εικόνα 3.13 Απεικόνιση του βρόχου συλλογής tweets

3.3. Προεπεξεργασία δεδομένων

Αφού ολοκληρωθεί η συγκέντρωση των δεδομένων, το επόμενο βήμα είναι η προεπεξεργασία τους. Το στάδιο αυτό αφορά στον καθαρισμό των πρωτογενών κειμένων από περιττά στοιχεία, είτε επειδή δεν προσφέρουν χρήσιμη πληροφορία στους μηχανισμούς που θα χρησιμοποιήσουμε (μοντέλα), είτε επειδή μπορεί να δημιουργήσουν προβλήματα στην διαδικασία εξαγωγής νοήματος. Υπάρχουν πολλά πιθανά βήματα επεξεργασίας που μπορούν να ακολουθηθούν, όπως ενδεικτικά:

- αφαίρεση υπερσυνδέσμων (URLs)
- μετατροπή HTML χαρακτήρων διαφυγής (escape characters)
- αντικατάσταση ή αφαίρεση emoticons και emojis (εικονιδίων που αποδίδουν συναίσθημα, είτε ως γραφικά, είτε ως μικρές ομάδες χαρακτήρων)
- αφαίρεση σημείων στίξεων και συμβόλων
- αφαίρεση αριθμών
- αφαίρεση τυχαίων χαρακτήρων / λέξεων ενός χαρακτήρα
- μετατροπή όλων των χαρακτήρων σε πεζούς
- διόρθωση ορθογραφικών λαθών
- αφαίρεση stop-words (λέξεων που θεωρείται ότι δεν φέρουν συναισθηματική πληροφορία)

- λημματοποίηση (μετατροπή των λέξεων σε μια πρωτογενή, ριζική μορφή - για παράδειγμα, ενικός, ονομαστική, χωρίς υποκοριστικά και άλλα)

Επιπλέον, πιο ειδικά για κείμενα που προέρχονται από το Twitter που μπορούμε να εφαρμόσουμε και βήματα όπως:

- αφαίρεση αναφορών σε ονόματα χρηστών (mentions)
- σπάσιμο φράσεων με κάτω παύλες (“ _ “ - underscores) στις επιμέρους λέξεις

Κάθε ένα από τα παραπάνω βήματα έχει συγκεκριμένη χρησιμότητα, και διαφορετικό τρόπο εφαρμογής. Πολλά από αυτά (όπως η αφαίρεση χαρακτήρων και η μετατροπή σε πεζά) μπορούν να εκτελεστούν ακόμα και με μία από τις εντολές που περιλαμβάνονται σε βιβλιοθήκες της Python, ενώ άλλα (όπως η διόρθωση ορθογραφικών λαθών και η αφαίρεση stop-words) μπορεί να απαιτούν την πρόσβαση σε λίστες λέξεων τρίτων βιβλιοθηκών που προορίζονται για επεξεργασία φυσικής γλώσσας (όπως η NLTK - Natural Language Toolkit). Επίσης, το ποια από αυτά τα βήματα θα επιλεγούν για εφαρμογή στο εκάστοτε dataset, εξαρτάται από την μεθοδολογία ανάλυσης που θα ακολουθηθεί στα επόμενα στάδια. Για παράδειγμα, η ανάλυση συναισθήματος με χρήση λεξικού απαιτεί τα περισσότερα από τα προαναφερθέντα βήματα προεπεξεργασίας, ώστε να είναι πιθανότερη η ταύτιση των εναπομεινουσών λέξεων με τους περιορισμένους αριθμητικά όρους που περιλαμβάνει ένα προαξιολογημένο συναισθηματικό λεξικό. Αντίθετα, η συναισθηματική ανάλυση μέσω ενός μοντέλου NLI που έχει προ-εκπαιδευτεί σε τεράστιο όγκο φυσικών κειμένων, πιθανότατα θα ωφεληθεί περισσότερο από μια πιο περιορισμένη προεπεξεργασία, που θα διατηρεί την φυσικότητα της γλώσσας των κειμένων του dataset.

Κατά το αρχικό φόρτωμα του dataset στην μνήμη, μπορούμε να υποδείξουμε τις στήλες που θέλουμε να συμπεριληφθούν (εν προκειμένω, μόνο αυτήν που περιέχει το κείμενο των tweets), όπως και ένα υποσύνολο των εγγραφών αν το dataset είναι πολύ μεγάλο για τις ανάγκες μας (για παράδειγμα τις πρώτες 20000).

3.4. Διερεύνηση επέκτασης

Όπως επισημάνθηκε στην περίληψη της διπλωματικής εργασίας, μετά την συλλογή και τον καθαρισμό του dataset από περιττά στοιχεία, το επόμενο στάδιο είναι η εξέταση της εφικτότητας βελτίωσης των δυνατοτήτων κατανόησης των χρησιμοποιούμενων μοντέλων, μέσω εμπλουτισμού των λεξικών τους με λέξεις από τα συλλεγμένα κείμενα.

Οι δύο συνηθέστερες μέθοδοι προσαρμογής ενός μοντέλου επεξεργασίας φυσικής γλώσσας (NLP) σε ένα νέο πεδίο κειμένων (για παράδειγμα, για λόγους εξειδίκευσης ή επιτέλεσης μιας νέας εργασίας) είναι:

- είτε η βελτιστοποίηση (fine-tuning) του προεκπαιδευμένου μοντέλου πάνω στα νέα κείμενα, που σημαίνει ότι δεν αλλάζει το λεξικό του tokenizer αλλά μόνο οι ενσωματώσεις (word embeddings - διανυσματικές αναπαραστάσεις του νοήματος) των λέξεων που ήδη γνωρίζει το μοντέλο, με αποτέλεσμα η απόδοση σε κείμενα του νέου πεδίου να μην είναι η ιδανική
- είτε η εξαρχής επανεκπαίδευση του μοντέλου πάνω σε νέα κείμενα, κάτι όμως που είναι απαιτητικό από πλευράς υπολογιστικού χρόνου και απαιτούμενου όγκου κειμένων του νέου πεδίου.

Μια ενδιάμεση λύση είναι η επέκταση του λεξικού του tokenizer ενός προεκπαιδευμένου μοντέλου με λέξεις από ένα νέο σώμα κειμένων, τεχνική που έχει χρησιμοποιηθεί στο μοντέλο exBERT (Tai et al, 2020). Το μοντέλο exBERT χρησιμοποιεί sub-word (επιπέδου υπολέξεων) tokenizer τύπου WordPiece - όμως, και ο tokenizer τύπου SentencePiece που χρησιμοποιείται στο μοντέλο DeBERTa που επιλέχθηκε για αυτήν την εργασία, μπορεί να δημιουργήσει tokens ολοκληρωμένων λέξεων στο τέλος του λεξικού του. Στο στάδιο αυτό παρατίθεται η μεθοδολογία για την εξαγωγή του νέου λεξικού από το dataset που δημιουργήθηκε στο προηγούμενο βήμα και η ενσωμάτωσή του στο μοντέλο, με μέρος του κώδικα να προέρχεται από τον (Guillou, 2021).

Για την δημιουργία του νέου λεξικού, γίνεται χρήση μίας ελεύθερης βιβλιοθήκης ανοιχτού κώδικα για διεργασίες επεξεργασίας φυσικής γλώσσας (NLP), της spaCY (*https://spacy.io/*). Μία μόνο από τις πολλές λειτουργίες της είναι αυτή του tokenization ολοκληρωμένων λέξεων μέσω αξιοποίησης των κενών εντός των προτάσεων, λαμβάνοντας υπόψην τις ιδιαιτερότητες κάθε γλώσσας πάνω στην οποία έχει προεκπαιδευτεί. Η πρώτη συνάρτηση του κώδικα αναλαμβάνει την αρχικοποίηση του tokenizer της spaCY (spacy_tokenizer) με τις ελληνικές παραμέτρους προεκπαίδευσης πάνω σε ειδησεογραφικά κείμενα (el_core_news_sm). Με την εφαρμογή της στο προεπεξεργασμένο dataset, κάθε εγγραφή (tweet) μετατρέπεται σε μία λίστα από tokens (αυτόνομες λέξεις).

Το επόμενο βήμα, είναι ο υπολογισμός της συχνότητας εμφάνισης των λέξεων-tokens μέσα στα νέα κείμενα ώστε, μέσω του καθορισμού των επιθυμητών ποσοστών εμφάνισης να επιτευχθεί μια ισορροπία μεταξύ σημαντικότητας των tokens και αύξησης του μεγέθους του λεξικού με νέους όρους. Για την λειτουργία αυτή, χρησιμοποιούμε την κλάση TfidfVectorizer της

βιβλιοθήκης ανάλυσης δεδομένων scikit-learn, η οποία αξιοποιεί τον spaCY tokenizer για να δημιουργήσει ένα πίνακα καταμέτρησης εμφανίσεων όρων, υπολογίζει τις αντίστροφες συχνότητές τους (IDF) και επιπλέον τις μετατρέπει σε μια κανονικοποιημένη διανυσματική αναπαράσταση της "αξίας" των όρων βάσει της τεχνικής TF-IDF *. Για τις ανάγκες του σταδίου αυτού, γίνεται χρήση μόνο των IDF τιμών που υπολογίζονται ως μέρος της παραπάνω διαδικασίας, ώστε μέσω συναρτήσεων να δημιουργηθεί μια λίστα όλων των tokens των νέων κειμένων, ταξινομημένων βάσει ποσοστιαίας συχνότητας εμφάνισης.

Επόμενο βήμα, αφού υπολογιστεί για εποπτικούς λόγους το μέγεθος της λίστας νέων όρων που δεν περιέχονται ήδη στο λεξικό του tokenizer, είναι η προσθήκη αυτών των όρων στον tokenizer, η επέκταση του πίνακα αναπαραστάσεων (embeddings matrix) του μοντέλου κατά το ίδιο μέγεθος, και η τοπική αποθήκευσή των εκτεταμένων εκδοχών. Το τελευταίο τμήμα κώδικα κάνει χρήση ενός μικρού κειμένου για να αξιολογήσει την απόδοση του tokenizer ως προς την αποδοτικότητα της κωδικοποίησης σε tokens (encoding) και την αποτελεσματικότητα της μετατροπής των tokens πίσω στο αρχικό κείμενο (decoding).

* Η τεχνική αντίστροφης συχνότητας όρων TF-IDF (Term frequency - Inverse document frequency) υπολογίζει των αριθμό εμφανίσεων κάθε όρου σε ένα σύνολο κειμένων, και κατόπιν χρησιμοποιεί αλγορίθμους για να αποδώσει μεγαλύτερη σημασία σε όρους που εμφανίζονται λιγότερο συχνά και άρα είναι πιθανότερο να περιέχουν πιο εξειδικευμένο ή ιδιαίτερο νόημα σε σχέση με άλλους που είναι πολύ κοινοί. Οι δύο παράγοντες που συνυπολογίζονται από τους αλγορίθμους είναι α) ο αριθμός εμφανίσεων της εκάστοτε λέξης σε ένα κείμενο και β) η αντίστροφη συχνότητα της λέξης στο σύνολο των κειμένων.

3.5. Ανάλυση συναισθήματος

Το τέταρτο στάδιο της διαδικασίας είναι η χρήση του μοντέλου μηχανικής μάθησης για την αναγνώριση των συναισθημάτων στα κείμενα του dataset (tweets) και η παραγωγή των σχετικών στατιστικών αποτελεσμάτων. Αρχικά, και αφού εισαχθούν οι απαραίτητες βιβλιοθήκες και αρχικοποιηθεί το μοντέλο, ορίζονται τα συναισθήματα που θέλουμε να εξεταστούν. Ο ορισμός αυτός υλοποιείται ως λίστα από υποθέσεις (hypothesis) που θα αξιολογηθούν μέσω της τεχνικής συμπερασμάτων φυσικής γλώσσας (NLI) και, για τις ανάγκες αυτής της εργασίας, περιλαμβάνει τέσσερα από τα πιο θεμελιώδη ανθρώπινα συναισθήματα - την χαρά, την λύπη, τον θυμό και τον φόβο.

Ακολούθως, αφού δημιουργηθεί ο κενός πίνακας αποθήκευσης των αποτελεσμάτων (MaxEnt, MinCont, BestGuess) υλοποιείται βρόχος που, διαδοχικά για κάθε tweet του dataset:

- υπολογίζει το ποσοστό συμφωνίας (entailment) και το ποσοστό αντίθεσης (contradiction) του κειμένου (ως premise) με κάθε ένα από τα συναισθήματα που

ορίσαμε αρχικά (ως hypothesis). Τα ποσοστά αυτά αποθηκεύονται σε αντίστοιχες λίστες συμφωνίας και αντίθεσης (lstent, lstcon).

- αποθηκεύει στον πίνακα αποτελεσμάτων, για το συγκεκριμένο tweet, το συναίσθημα με το μεγαλύτερο ποσοστό συμφωνίας (MaxEnt) και το συναίσθημα με το μικρότερο ποσοστό αντίθεσης (MinCont).
- υπολογίζει έναν "δείκτη αβεβαιότητας" του μοντέλου για το συναίσθημα κάθε μίας από τις προηγούμενες δύο προβλέψεις (MaxEnt και MinCont), και αποθηκεύει στον πίνακα αποτελεσμάτων, για το συγκεκριμένο tweet, το συναίσθημα με την χαμηλότερη τιμή αβεβαιότητας (BestGuess). Το σκεπτικό αυτής της πρότασης είναι να επηρεάσει το αποτέλεσμα και η αντίθεση που μετράται για ένα συναίσθημα, σε περίπτωση που η συμφωνία δεν παρέχει από μόνη της ένα αποτέλεσμα υψηλής αυτοπεποίθησης. Ο δείκτης αυτός μπορεί να υπολογιστεί για κάθε συναίσθημα X ως εξής:
  (απόσταση entailment X από το ιδανικό 100%) + (απόσταση contradiction X από το ιδανικό 0%)
- τυπώνει μήνυμα ολοκλήρωσης της αξιολόγησης του συγκεκριμένου tweet.

Αφού ο βρόχος ολοκληρώσει την επεξεργασία όλων των tweets του dataset, το τελευταίο τμήμα του κώδικα υπολογίζει και τυπώνει έναν πίνακα με στατιστικά στοιχεία για την συνολική ποσοστιαία εμφάνιση κάθε συναισθήματος στα tweets του dataset, βάσει κάθε μίας από τις τεχνικές αξιολόγησης - μέγιστης συμφωνίας (MaxEnt), ελάχιστης αντίθεσης (MinCont) και μικρότερου δείκτη αβεβαιότητας (BestGuess) - ώστε να είναι εφικτή και η σύγκριση των αποτελεσμάτων των τεχνικών αυτών.

4. Συλλογή και αξιολόγηση αποτελεσμάτων

Στο κεφάλαιο αυτό παρουσιάζεται η εκτέλεση του κώδικα που περιγράφηκε νωρίτερα, για την συλλογή των δεδομένων, την επεξεργασία τους, την ανάλυση συναισθημάτων και την παρουσίαση των αποτελεσμάτων. Αρχικά, για τις ανάγκες των επόμενων βημάτων της εργασίας, συλλέχθηκαν τα παρακάτω datasets σε διαφορετικές χρονικές περιόδους - όλα τα μηνύματα ζητήθηκε από το API του Twitter να έχουν κύρια γλώσσα τα ελληνικά, ενώ έχουν εξαιρεθεί από την αναζήτηση τα αναμοιρασμένα μηνύματα (retweets):

- το PoliticalTweets.csv αποτελείται από tweets που συλλέχθηκαν σε δύο συνεχόμενα χρονικά διαστήματα μεταξύ 6 και 17 Φεβρουαρίου 2023, με αναφορές σε εγχώρια πολιτικά πρόσωπα και παρατάξεις και αριθμεί 63.000 μηνύματα. Το ερώτημα που χρησιμοποιήθηκε ήταν το:

Μητσοτακη OR Τσιπρα OR ΝΔ OR συριζα OR εκλογ OR βουλη OR βουλευτ OR κιναλ OR ανδρουλακη OR κκε OR κουτσουμπα OR βελοπουλ OR βαρουφακη OR μερα25

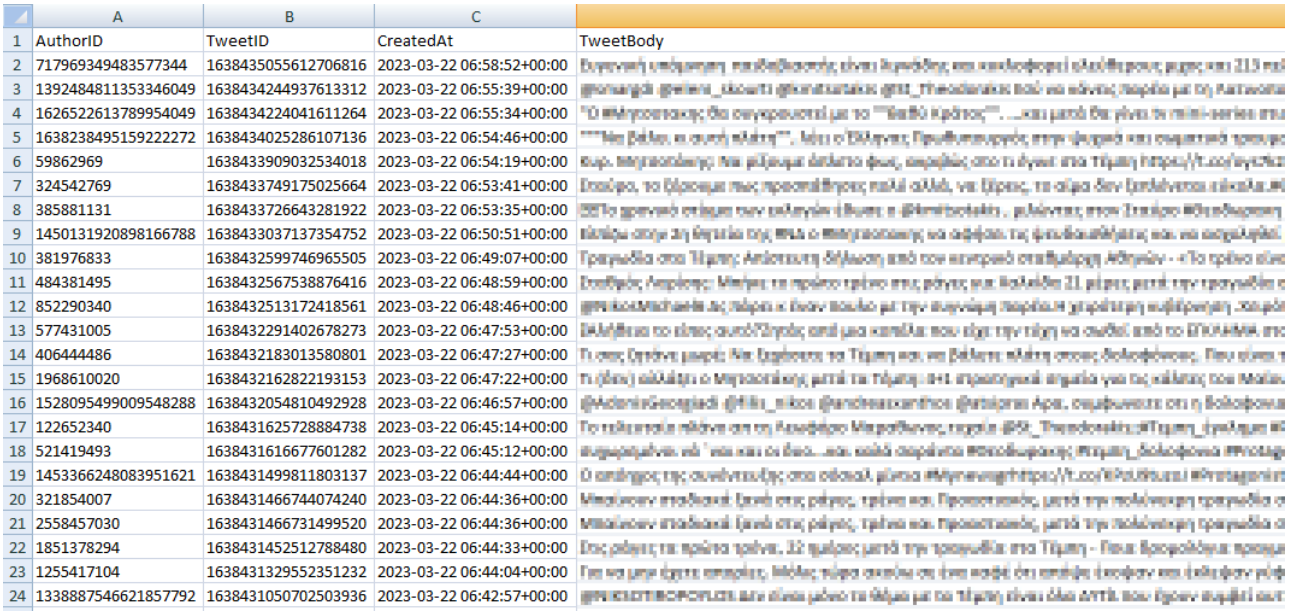
- το TempiTweets.csv αποτελείται από tweets που δημιουργήθηκαν τις πρώτες επτά ημέρες (1 έως 7 Μαρτίου) μετά το σιδηροδρομικό δυστύχημα της 28ης Φεβρουαρίου 2023 στα Τέμπη και περιλαμβάνει 101.000 μηνύματα. Το ερώτημα που χρησιμοποιήθηκε ήταν το:
  Τεμπη OR Λαρισα OR τρενο

- και το μικρότερο SpitiMegaTweets.csv που αποτελείται από περίπου 1100 tweets που χρησιμοποιούν την θεματική επισήμανση δημοφιλούς τηλεοπτικής ψυχαγωγικής εκπομπής και συλλέχθηκαν κατά την ώρα προβολής της αλλά και τις ακόλουθες 12 ώρες, το Σάββατο 1 και την Κυριακή 2 Απριλίου 2023. Το ερώτημα που χρησιμοποιήθηκε ήταν το:
  spitimetomega

Στις επόμενες εικόνες παρουσιάζεται μια τυπική έξοδος που τυπώνεται από τον κώδικα κατά την διάρκεια της συλλογής και αποθήκευσης των μηνυμάτων, καθώς και ένα δείγμα των ανεπεξέργαστων datasets που δημιουργούνται από την διαδικασία αυτή.

```
*** Κωδικός απάντησης API: 200
Περισσότερα tweets διαθέσιμα. Token επόμενης σελίδας:  b26v89c19zqg8o3fqka29k9555hwwczpzd8n49kgs2m4d
Χρονικό διάστημα (path 1) από:  2023-03-21T22:00:00.000Z  έως:  2023-03-22T07:00:00.000Z
Σελίδα: 1
# από tweets που έχουν ληφθεί ως τώρα (path 1):  100


*** Κωδικός απάντησης API: 200
Περισσότερα tweets διαθέσιμα. Token επόμενης σελίδας:  b26v89c19zqg8o3fqka29i6t23iv6523u1lrxvb7v9jlp
Χρονικό διάστημα (path 2) από:  2023-03-21T22:00:00.000Z  έως:  2023-03-22T07:00:00.000Z
Σελίδα: 2
# από tweets που έχουν ληφθεί ως τώρα (path 2):  200

                                        •••

*** Κωδικός απάντησης API: 200
Περισσότερα tweets διαθέσιμα. Token επόμενης σελίδας:  b26v89c19zqg8o3fqka29i43v7p00e6tecwldjl93dbb1
Χρονικό διάστημα (path 2) από:  2023-03-21T22:00:00.000Z  έως:  2023-03-22T07:00:00.000Z
Σελίδα: 6
# από tweets που έχουν ληφθεί ως τώρα (path 2):  600


*** Κωδικός απάντησης API: 200
Δεν υπάρχουν άλλα tweets. Τελευταία σελίδα για αυτό το χρονικό διάστημα.
Χρονικό διάστημα (path 3) από:  2023-03-21T22:00:00.000Z  έως:  2023-03-22T07:00:00.000Z
Page: 7
# από tweets που έχουν ληφθεί ως τώρα (path 3):  629


---------------------------------------
---------------------------------------


Τέλος. Συνολικός αριθμός tweets που αποθηκεύτηκαν στο .csv αρχείο:  629
```

Εικόνα 4.1΄Εξοδος κώδικα συλλογής tweets



	A	B	C	TweetBody
1	AuthorID	TweetID	CreatedAt	TweetBody
2	717969349483577344	1638435055612706816	2023-03-22 06:58:52+00:00	[redacted]
3	1392484811353346049	1638434244937613312	2023-03-22 06:55:39+00:00	[redacted]
4	1626522613789954049	1638434224041611264	2023-03-22 06:55:34+00:00	[redacted]
5	1638238495159222272	1638434025286107136	2023-03-22 06:54:46+00:00	[redacted]
6	59862969	1638433909032534018	2023-03-22 06:54:19+00:00	[redacted]
7	324542769	1638433749175025664	2023-03-22 06:53:41+00:00	[redacted]
8	385881131	1638433726643281922	2023-03-22 06:53:35+00:00	[redacted]
9	1450131920898166788	1638433037137354752	2023-03-22 06:50:51+00:00	[redacted]
10	381976833	1638432599746965505	2023-03-22 06:49:07+00:00	[redacted]
11	484381495	1638432567538876416	2023-03-22 06:48:59+00:00	[redacted]
12	852290340	1638432513172418561	2023-03-22 06:48:46+00:00	[redacted]
13	577431005	1638432291402678273	2023-03-22 06:47:53+00:00	[redacted]
14	406444486	1638432183013580801	2023-03-22 06:47:27+00:00	[redacted]
15	1968610020	1638432162822193153	2023-03-22 06:47:22+00:00	[redacted]
16	1528095499009548288	1638432054810492928	2023-03-22 06:46:57+00:00	[redacted]
17	122652340	1638431625728884738	2023-03-22 06:45:14+00:00	[redacted]
18	521419493	1638431616677601282	2023-03-22 06:45:12+00:00	[redacted]
19	1453366248083951621	1638431499811803137	2023-03-22 06:44:44+00:00	[redacted]
20	321854007	1638431466744074240	2023-03-22 06:44:36+00:00	[redacted]
21	2558457030	1638431466731499520	2023-03-22 06:44:36+00:00	[redacted]
22	1851378294	1638431452512788480	2023-03-22 06:44:33+00:00	[redacted]
23	1255417104	1638431329552351232	2023-03-22 06:44:04+00:00	[redacted]
24	1338887546621857792	1638431050702503936	2023-03-22 06:42:57+00:00	[redacted]

Εικόνα 4.2 Τμήμα πρωτογενούς dataset

Το επόμενο βήμα είναι ο καθαρισμός των datasets. Για κάθε ένα από αυτά, διενεργούμε την προεπεξεργασία με δύο διαφορετικές προσεγγίσεις: α) εφαρμόζοντας όλα τα διαθέσιμα στάδια καθαρισμού, όπως πιθανόν θα ήταν ορθότερο αν τα επεξεργασμένα κείμενα προοριζόντουσαν για ανίχνευση συναισθήματος με χρήση λεξικών (στην περίπτωση αυτή, τα προκύπτοντα datasets ονομάζονται βάσει των αρχικών με την προσθήκη του *-Cleaned-All*) και β) εφαρμόζοντας ένα πιο περιορισμένο αριθμό σταδίων καθαρισμού, που θα επιτρέψει την διατήρηση μιας πιο φυσικής μορφής των κειμένων (κατά την προσέγγιση αυτή, τα προκύπτοντα datasets ονομάζονται βάσει των αρχικών με την προσθήκη του *-Cleaned-Part*). Κατά το φόρτωμα του πρωτογενούς dataset υπάρχει η επιλογή ανάγνωσης και επεξεργασίας μόνο ενός υποσυνόλου των δεδομένων (αν το dataset είναι πολύ μεγάλο) αλλά, σε κάθε περίπτωση, πριν τον καθαρισμό διατηρείται μόνο η στήλη του κυρίως κειμένου των tweets και απορρίπτονται οι υπόλοιπες πληροφορίες (αναγνωριστικά χρήστη-μηνύματος και ημερομηνία-ώρα) που δεν συνεισφέρουν στην εκπαίδευση μοντέλων και στην συναισθηματική ανάλυση.

Για την δεύτερη προσέγγιση, του καθαρισμού με σκοπό την διατήρηση μιας φυσικότερης μορφής των αρχικών κειμένων, επιλέχθηκε από τα διαθέσιμα στον κώδικα στάδια να χρησιμοποιηθούν τα εξής:

- αφαίρεσης υπερσυνδέσμων (links)
- μετατροπής HTML χαρακτήρων διαφυγής
- αφαίρεσης αναφορών σε χρήστες (mentions)
- σπάσιμου φράσεων με underscores
- αφαίρεσης emoticons

- αφαίρεση εισαγωγικών και hashtags

και να παραλειφθούν τα στάδια:

- αφαίρεσης σημείων στίξης και συμβόλων (πλην εισαγωγικών και hashtags)
- αφαίρεσης αριθμών και τυχόν υπόλοιπων μη αλφαβητικών χαρακτήρων
- αφαίρεσης λέξεων ενός χαρακτήρα
- μετατροπή όλων των χαρακτήρων σε πεζούς
- διόρθωσης ορθογραφικών λαθών
- αφαίρεσης stop-words
- λημματοποίησης

Με την ολοκλήρωση της επεξεργασίας, ο κώδικας τυπώνει συγκριτικά την αρχική και τελική μορφή των κειμένων (καθώς και ενδιάμεσες αν το επιλέξουμε), ώστε να είναι εμφανής η λειτουργικότητα των σταδίων και να εντοπιστούν ευκολότερα τυχόν προβλήματα ή λαθη. Ακολουθούν δύο εικόνες από τις αλλαγές που πραγματοποιεί ο κώδικας στα πρωτογενή κείμενα, κατά τις δύο διαφορετικές προσεγγίσεις καθαρισμού.

Out[15]:

	TweetBody	Cleaned	Tokenized	Final
0	Πλειστηριασμοί: Έργο των «καραγκιόζηδων» του Σ...	πλειστηριασμοί ργο των καραγκιόζηδων του συριζ...	[πλειστηριασμοί, έργο, καραγκιόζηδων, συριζα, ...	πλειστηριασμοί έργο καραγκιόζηδων συριζα τώρα ...
1	Τα ραντεβού Μητσοτάκη στη Διάσκεψη Ασφαλείας τ...	τα ραντεβού μητσοτάκη στη διάσκεψη ασφαλείας τ...	[ραντεβού, μητσοτάκη, διάσκεψη, ασφαλείας, μον...	ραντεβού μητσοτάκη διάσκεψη ασφαλείας μονάχους
2	@lykomhtros @zeyele Απο τη στιγμη που ζητας απ...	απο τη στιγμη που ζητας απο ενα κομμουνιστικο ...	[τη, στιγμη, ζητάς, ενα, κομμουνιστικό, κομμα,...	τη στιγμη ζητάς ενα κομμουνιστικό κομμα καταργ...
3	@NikosAnevlavis @AnnaAsimakopoul @Europarl_EN ...	αν το που ψήφισε νδ δεν είναι πλειοψηφία το πο...	[ψήφισε, νδ, δεν, είναι, πλειοψηφία, έβγαλε, σ...	ψήφισε νδ δεν είναι πλειοψηφία έβγαλε συριζα ήταν
4	Το παρασκήνιο της συνάντησης Τσίπρα - Στουρνάρ...	το παρασκήνιο της συνάντησης τσίπρα στουρνάρα ...	[παρασκήνιο, της, συνάντησης, τσίπρα, στουρνάρ...	παρασκήνιο της συνάντησης τσίπρα στουρνάρα σα ...
5	@velzevoul1 @atsipras Βρες δουλειά τουρκόσπορε...	βρες δουλειά τουρκόσπορε αυτά δεν θα πουλάνε σ...	[βρες, δουλειά, τουρκόσπορε, αυτά, δεν, πουλάν...	βρες δουλειά τουρκόσπορε αυτά δεν πουλάνε μήνε...
6	Ισπανία: Η Βουλή ενέκρινε νόμο που προβλέπει -...	ισπανία βουλή ενέκρινε νόμο που προβλέπει άδει...	[ισπανία, βουλή, ενέκρινε, νόμο, προβλέπει, άδ...	ισπανία βουλή ενέκρινε νόμο προβλέπει άδεια εμ...
7	Μανούλες του fb και του twitter ότι πιο ηλίθιο...	μανούλες του fb και του twitter ότι πιο ηλίθιο...	[μανούλες, fb, twitter, ότι, πιο, ηλίθιο, κυκλ...	μανούλες fb twitter ότι πιο ηλίθιο κυκλοφορεί ...
8	@gmkpapakostas @LeoKosmas Χαίρομαι που καταλαβ...	χαίρομαι που καταλαβαίνετε τη χρησιμότητα του ...	[χαίρομαι, καταλαβαίνετε, τη, χρησιμότητα, συρ...	χαίρομαι καταλαβαίνετε τη χρησιμότητα συριζα ω...
9	Ψήνεται και νέα αγορά Rafale; Στο ΓΕΑ δεν μυρί...	ψήνεται και νέα αγορά rafale στο γεα δεν μυρίζ...	[ψήνεται, νέα, αγορά, rafael, δεν, μυρίζει, κά...	ψήνεται νέα αγορά rafael δεν μυρίζει κάτι τέτο...
10	@vpng13 Γιατί, πιστεύεις πως υπάρχει έστω και ...	γιατί πιστεύεις πως υπάρχει έστω και ένας δεξί...	[γιατί, πιστεύεις, υπάρχει, έστω, ένας, δεξιός...	γιατί πιστεύεις υπάρχει έστω ένας δεξιός ενοχλ...
11	..κι όχι πώς κακαρίζουν οι κότες του ΣΥΡΙΖΑ στ...	κι όχι πώς κακαρίζουν οι κότες του συριζα στου...	[όχι, πώς, καθαρίζουν, κότες, συριζα, στους, δ...	όχι πώς καθαρίζουν κότες συριζα στους δρόμους
12	@MariaKappatou @AretiAthanasiu Το μποϊκοτάζ στ...	το μποϊκοτάζ στα γαλακτοκομικά δεν το πρότεινε...	[μποϊκοτάζ, στα, γαλακτοκομικά, δεν, πρότεινε,...	μποϊκοτάζ στα γαλακτοκομικά δεν πρότεινε ουτε ...
13	@mulina_cz @w1qRzns6NiIMv9e Δυστυχώς υπάρχουν ...	δυστυχώς υπάρχουν βόδια που δεν καταλαβαίνουν ...	[δυστυχώς, υπάρχουν, βόδια, δεν, καταλαβαίνουν...	δυστυχώς υπάρχουν βόδια δεν καταλαβαίνουν λένε...
14	Τσιπρας θα πάει στις εκλογές με ψεύδη, λάστη κ...	τσιπρας θα πάει στις εκλογές με ψεύδη λάστη κα...	[τσιπρας, πάει, στις, εκλογές, ψεύδη, λάστη, σ...	τσιπρας πάει στις εκλογές ψεύδη λάστη στέρνοντ...
15	Το τραγούδι του τίμιου βουλευτή της ΕΡΕ:"Δυο π...	το τραγούδι του τίμιου βουλευτή της ερε δυο πό...	[τραγούδι, τίμιος, βουλευτή, της, ερε, δυο, πό...	τραγούδι τίμιος βουλευτή της ερε δυο πόρτες έχ...
16	Περιοδεία Τσίπρα σε Τρίκαλα και Καρδίτσα - Πλε...	περιοδεία τσίπρα σε τρίκαλα και καρδίτσα πλεισ...	[περιοδεία, τσίπρα, τρίκαλα, καρφίτσα, πλειστη...	περιοδεία τσίπρα τρίκαλα καρφίτσα πλειστηριασμ...
17	@arhontas1 @nikos_sverkos εσύ πλήρωνε τα δάνει...	εσύ πλήρωνε τα δάνεια πασοκ νδ και μη λες πολλά	[εσύ, πλήρωνε, δάνεια, πασοκ, νδ, λες, πολλά]	εσύ πλήρωνε δάνεια πασοκ νδ λες πολλά
18	Μνημονιακη υποχρέωση ήταν (εσείς χρεοκοπησατε...	μνημονιακη υποχρέωση ήταν εσείς χρεοκοπησατε τ...	[μνημονιακη, υποχρέωση, ήταν, εσείς, χρεοκοπησ...	μνημονιακη υποχρέωση ήταν εσείς χρεοκοπησατε τ...
19	Τοξικό πολιτικό κλίμα - Γιατί ο Μητσοτάκης ισχ...	τοξικό πολιτικό κλίμα γιατί μητσοτάκης ισχυρίζ...	[τοξικό, πολιτικό, κλίμα, γιατί, μητσοτάκης, ι...	τοξικό πολιτικό κλίμα γιατί μητσοτάκης ισχυρίζ...

Εικόνα 4.3 Εκτεταμένος καθαρισμός εγγραφών του dataset

Out[8]:

	TweetBody	Cleaned
0	Πλειστηριασμοί: Έργο των «καραγκιόζηδων» του Σ...	Πλειστηριασμοί: Έργο των καραγκιόζηδων του ΣΥΡ...
1	Τα ραντεβού Μητσοτάκη στη Διάσκεψη Ασφαλείας τ...	Τα ραντεβού Μητσοτάκη στη Διάσκεψη Ασφαλείας τ...
2	@lykomhtros @zeyele Απο τη στιγμη που ζητας απ...	Απο τη στιγμη που ζητας απο ενα κομμουνιστικ...
3	@NikosAnevlavis @AnnaAsimakopoul @Europarl_EN ...	Αν το 40 που ψήφισε ΝΔ δεν είναι πλειοψηφία...
4	Το παρασκήνιο της συνάντησης Τσίπρα - Στουρνάρ...	Το παρασκήνιο της συνάντησης Τσίπρα - Στουρνάρ...
5	@velzevoul1 @atsipras Βρες δουλειά τουρκόσπορε...	Βρες δουλειά τουρκόσπορε. Αυτά δεν θα πουλάν...
6	Ισπανία: Η Βουλή ενέκρινε νόμο που προβλέπει -...	Ισπανία: Η Βουλή ενέκρινε νόμο που προβλέπει -...
7	Μανούλες του fb και του twitter ότι πιο ηλίθιο...	Μανούλες του fb και του twitter ότι πιο ηλίθιο...
8	@gmkpapakostas @LeoKosmas Χαίρομαι που καταλαβ...	Χαίρομαι που καταλαβαίνετε τη χρησιμότητα το...
9	Ψήνεται και νέα αγορά Rafale; Στο ΓΕΑ δεν μυρί...	Ψήνεται και νέα αγορά Rafale; Στο ΓΕΑ δεν μυρί...
10	@vpng13 Γιατί, πιστεύεις πως υπάρχει έστω και ...	Γιατί πιστεύεις πως υπάρχει έστω και ένας δεξ...
11	..κι όχι πώς κακαρίζουν οι κότες του ΣΥΡΙΖΑ στ...	..κι όχι πώς κακαρίζουν οι κότες του ΣΥΡΙΖΑ στ...
12	@MariaKappatou @AretiAthanasiu Το μποϊκοτάζ στ...	Το μποϊκοτάζ στα γαλακτοκομικά δεν το πρότει...
13	@mulina_cz @w1qRzns6NilMv9e Δυστυχώς υπάρχουν ...	Δυστυχώς υπάρχουν βόδια που δεν καταλαβαίνου...
14	Τσιπρας θα πάει στις εκλογές με ψεύδη, λάσπη κ...	Τσιπρας θα πάει στις εκλογές με ψεύδη λάσπη κα...
15	Το τραγούδι του τίμιου βουλευτή της ΕΡΕ:"Δυο π...	Το τραγούδι του τίμιου βουλευτή της ΕΡΕ:Δυο πό...
16	Περιοδεία Τσίπρα σε Τρίκαλα και Καρδίτσα - Πλε...	Περιοδεία Τσίπρα σε Τρίκαλα και Καρδίτσα - Πλε...

Εικόνα 4.4 Περιορισμένος καθαρισμός εγγραφών του dataset

Στην περίπτωση πρωτογενών datasets των 50-100.000 tweets και κάνοντας χρήση ενός οικιακού υπολογιστή μέτριας ισχύος (i7 7ης γενιάς, 16GB RAM), η επεξεργασία ολοκληρώνεται σε χρόνο ενός έως δύο λεπτών όταν εφαρμόζονται όλα τα διαθέσιμα στάδια, ενώ πολύ συντομότερα (εντός μερικών δευτερολέπτων) όταν παραλείπονται οι πιο χρονοβόρες διεργασίες όπως η εύρεση και διόρθωση ορθογραφικών λαθών.

Εξετάζοντας τα παραγόμενα datasets, επιβεβαιώνεται ότι η δεύτερη προσέγγιση με τα λιγότερα στάδια επεξεργασίας διατηρεί σε μεγάλο βαθμό την φυσικότητα της εκφοράς του γραπτού λόγου, σε αντίθεση με την πρώτη προσέγγιση που ουσιαστικά δημιουργεί λίστες λέξεων με μεγαλύτερη πιθανότητα εμφάνισης σε ένα υπάρχον συναισθηματικό λεξικό. Παρατηρούμε επίσης ότι:

- το στάδιο διόρθωσης ορθογραφικών λαθών σε κάποιες περιπτώσεις αντιλαμβάνεται και διορθώνει ως λάθη, λέξεις που πιθανόν δεν αναγνωρίζει - με επακόλουθη την αλλοίωση του νοήματος κάποιων μηνυμάτων (για παράδειγμα Τέμπη -> Τέμπο, Άρειος -> αέριος, Rafale -> Rafael)
- η αφαίρεση των stop-words σε πολλές περιπτώσεις αλλοιώνει το νόημα των προτάσεων, παρόλο που ως μεμονωμένες λέξεις όντως δεν φέρουν συναισθηματική πληροφορία και άρα δεν θα περιέχονται σε ένα προαξιολογημένο συναισθηματικό λεξικό. Άρα, όταν το επεξεργασμένο dataset προορίζεται για χρήση σε μοντέλα μηχανικής μάθησης, αν δεν παραλειφθεί εντελώς αυτό το στάδιο επεξεργασίας,

χρειάζεται προσοχή ώστε να εξαιρεθούν από την αφαίρεση λέξεις που επηρεάζουν το νόημα της φράσης (όπως τα αρνητικά μόρια “δε/δεν/μη/μην”).

- το στάδιο λημματοποίησης της βιβλιοθήκης NLTK δεν επιφέρει προς το παρόν κάποιες αλλαγές πάνω σε ελληνικά κείμενα.

Όπως παρουσιάστηκε στο προηγούμενο κεφάλαιο, μετά τον καθαρισμό των dataset ακολουθεί η δημιουργία λίστας λέξεων για τον εμπλουτισμό του λεξικού του συνδυασμού tokenizer-μοντέλου που επιλέξαμε (mDeBERTa). Χρησιμοποιώντας ως πηγή κειμένων το dataset των επεξεργασμένων πολιτικών tweets (PoliticalTweets-Cleaned-Part.csv), ο TfidfVectorizer μέσω του spaCY tokenizer παράγει περίπου 128.000 διαφορετικές λέξεις-tokens μέσα από ένα σύνολο 63.000 μηνυμάτων.

```
# Αποτέλεσμα:
# αριθμός προτάσεων-tweets / αριθμός συνολικών διαφορετικών λέξεων-tokens στο dataset
```

```
(63223, 128062)
CPU times: total: 3min 56s
Wall time: 3min 57s
```

Εικόνα 4.5 Αριθμός λέξεων από το tokenization μέσω spaCY

Στον πίνακα result αποθηκεύεται, για κάθε μήνυμα-tweet, μια λίστα με τις αριθμητικές αναπαραστάσεις κάθε λέξης του μηνύματος και η TF-IDF αξιολόγηση της λέξης. Ενδεικτικά, αν τυπώσουμε το περιεχόμενο του πίνακα για ένα από αυτά τα μηνύματα (για παράδειγμα το 5ο “*Το παρασκήνιο της συνάντησης Τσίπρα - Στουρνάρα: Όσα δεν ειπώθηκαν*” με την αρίθμηση να ξεκινάει από το 0), το αποτέλεσμα φαίνεται στην ακόλουθη εικόνα.

```
In [7]: print(result[4])
```

```
  (0, 68245)    0.4924278408166833
  (0, 10330)    0.3968262918579344
  (0, 39933)    0.34267983898254356
  (0, 309)      0.1718845792776498
  (0, 42245)    0.1927867812456783
  (0, 113667)   0.41062996348540626
  (0, 100705)   0.40386067603051923
  (0, 41772)    0.1856481150735459
  (0, 63295)    0.12929135540145723
  (0, 117111)   0.12583267348618254
  (0, 4917)     0.13102457988039515
```

Εικόνα 4.6 Καταχώρηση του πίνακα αποτελεσμάτων του TfidfVectorizer

Κατόπιν, χρησιμοποιώντας τις αντίστροφες συχνότητες που υπολογίστηκαν με την παραπάνω διαδικασία για τα tokens του νέου dataset, ο κώδικας δημιουργεί πίνακες που περιέχουν τα νέα

tokens ταξινομημένα βάσει του απόλυτου αριθμού εμφανίσεών τους μέσα στα tweets του dataset, αλλά και την ποσοστιαία συχνότητά τους. Οι επόμενες εικόνες εμφανίζουν το αρχικό τμήμα με τις πρώτες καταχωρήσεις αυτών των πινάκων.

```
In [45]: tokens_dfreqs

Out[45]: {'και': 24956,
          '.': 23982,
          'το': 20464,
          'να': 19654,
          'του': 19644,
          'ο': 17703,
          'για': 17246,
          'την': 14622,
          'με': 14607,
          'που': 14578,
          'ΝΔ': 13362,
          'η': 12707,
          'της': 12479,
          'τον': 12316,
          'ΣΥΡΙΖΑ': 12129,
          'θα': 11985,
          'τα': 11943,
          'δεν': 11612,
```

Εικόνα 4.7 Tokens και αριθμός εμφανίσεων

```
In [49]: pct_list

Out[49]: [39.472976607,
          37.93239802,
          32.367967354,
          31.086788036,
          31.070971007,
          28.000885754,
          27.278047546,
          23.127659238,
          23.103933695,
          23.058064312,
          21.134713633,
          20.098698259,
          19.738070006,
          19.48025244,
          19.184474005,
          18.956708793,
          18.890277273,
          18.366733625,
```

Εικόνα 4.8 Ποσοστιαίες συχνότητες tokens

Έχοντας πλέον μια λίστα των tokens από τα νέα κείμενα, ταξινομημένα βάσει της συχνότητας εμφάνισής τους, μπορούμε να επιλέξουμε ένα υποσύνολο αυτών για την επέκταση του λεξικού του μοντέλου. Για σκοπούς σύγκρισης με την αρχική μορφή του μοντέλου, επιλέχθηκαν 3 διαφορετικά υποσύνολα των tokens, βάσει ποσοστού εμφάνισής τους - μεγαλύτερο του 1%, μεγαλύτερο του 0,01% και μεγαλύτερο του 0% (δηλαδή συμπερίληψη όλων των tokens του dataset) και αποθηκεύτηκαν τα προκύπτοντα μοντέλα ως Extended1DeBERTa, Extended001DeBERTa, ExtendedAllDeBERTa. Στις παρακάτω εικόνες εμφανίζονται τα αποτελέσματα για κάθε μία από αυτές τις περιπτώσεις:

```
Υπάρχουν ήδη στο αρχικό / καινούριες, δεν υπάρχουν στο αρχικό / συνολικές στο νέο λεξικό:

(93, 96, 189)

[ BEFORE ] tokenizer vocab size: 250102
[ AFTER ] tokenizer vocab size: 250198

added_tokens: 96

Embedding(250198, 768)
```

Εικόνα 4.9 Καταμέτρηση-προσθήκη tokens συχνότητας >1%

```
Υπάρχουν ήδη στο αρχικό / καινούριες, δεν υπάρχουν στο αρχικό / συνολικές στο νέο λεξικό:
(883, 12780, 13663)

[ BEFORE ] tokenizer vocab size: 250102
[ AFTER ] tokenizer vocab size: 262882

added_tokens: 12780

Embedding(262882, 768)
```

Εικόνα 4.10 Καταμέτρηση-προσθήκη tokens συχνότητας >0,01%

```
Υπάρχουν ήδη στο αρχικό / καινούριες, δεν υπάρχουν στο αρχικό / συνολικές στο νέο λεξικό:
(3494, 124568, 128062)

[ BEFORE ] tokenizer vocab size: 250102
[ AFTER ] tokenizer vocab size: 374670

added_tokens: 124568

Embedding(374670, 768)
```

Εικόνα 4.11 Καταμέτρηση-προσθήκη συνόλου νέων tokens

Για την πραγματοποίηση της σύγκρισης του αρχικού, εκπαιδευμένου πολυγλωσσικού μοντέλου mDeBERTa-v3-base-mnli-xnli με τις τρεις νέες εκτεταμένες εκδοχές του, επιλέξαμε να χρησιμοποιήσουμε δύο μικρά κείμενα:

- το TestText1, το οποίο δημιουργήθηκε ως άθροισμα από tweets που προέρχονται από το dataset PoliticalTweets-Cleaned-Part.csv το οποίο χρησιμοποιήθηκε νωρίτερα, άρα αναμένεται το μεγαλύτερο μέρος των λέξεών του να περιλαμβάνεται στο λεξικό των εκτεταμένων μοντέλων.
- το TestText2, το οποίο αποτελεί παράγραφο άρθρου γενικότερης πολιτικής θεματολογίας από ελληνικό ειδησεογραφικό ιστότοπο, και άρα αποτελεί ένα πιο πιθανό δείγμα των τυχαίων κειμένων που θα κληθούν να διαχειριστούν τα μοντέλα.

Εκτελώντας το encoding (tokenization) του δεύτερου κειμένου (TestText2) μέσω του απείραχτου προεκπαιδευμένου μοντέλου (mDeBERTa-v3-base-mnli-xnli), ο κώδικας παράγει την ακόλουθη έξοδο:

```
Αριθμός tokens => 165

['_Το', '_νομ', 'οσχ', 'έ', 'διο', '_για', '_πρ', 'ώτη', '_', 'φορά', '_ανα', 'φέρ', 'εται', '_συ', 'γκεκριμέν', 'α', '_σ', 'το
υς', '_παρ', 'ό', 'χους', '_', 'υπηρε', 'σιών', '_', 'ύδ', 'ατος', '_', 'ως', '_δημ', 'όσι', 'ους', '_και', '_δημο', 'τικούς',
'_οργαν', 'ισμούς', ',', '_', 'δεν', '_', 'ασχολ', 'είται', '_σε', '_καν', 'ένα', '_άρθρο', '_του', '_με', '_το', '_ιδιο', 'κτ
η', 'σιακ', 'ό', '_καθ', 'εστ', 'ώς', ',', '_', 'ού', 'τε', '_με', '_τη', '_με', 'τοχ', 'ική', '_σύν', 'θεση', '_', 'των', '_πα
ρ', 'ό', 'χων', '_', 'υπηρε', 'σιών', '_', 'ύδ', 'ατος', ',', '_', 'η', '_', 'νέ', 'α', '_Ρ', 'υθ', 'μ', 'ιστική', '_Αρχ', 'ή',
'_', 'έχει', '_καθαρ', 'ά', '_επ', 'οπτικ', 'ές', '_και', '_γνω', 'μ', 'οδο', 'τικές', '_', 'αρμο', 'δι', 'ότητες', ',', '_',
'ενώ', '_οι', '_', 'κανον', 'ιστικές', '_', 'αρμο', 'δι', 'ότητες', '_για', '_', 'την', '_κ', 'οστ', 'ολόγηση', '_του', '_νε',
'ρού', '_παρα', 'μένου', 'ν', '_', 'αρμο', 'δι', 'ότητα', '_', 'των', '_', 'υπουργ', 'ών', '_και', '_για', '_', 'τον', '_', 'σκ
οπ', 'ό', '_αυτό', 'ν', '_εκδ', 'ίδ', 'ουν', '_', 'ΚΥ', 'Α', ',', '_εί', 'πε', '_', 'ο', '_κ', '.', '_Καπ', 'πάτ', 'ος', '.']

{'input_ids': [1, 3442, 30988, 166768, 1863, 17586, 902, 3787, 25189, 260, 8650, 3477, 13255, 5410, 10454, 28612, 682, 928, 937
9, 7126, 931, 64824, 260, 24190, 70151, 260, 232910, 154208, 260, 2047, 40000, 55941, 3309, 519, 54871, 119707, 47386, 79029, 2
62, 260, 2031, 260, 65279, 14796, 1262, 14370, 2218, 36208, 695, 851, 700, 84188, 46106, 129572, 931, 6335, 43595, 4337, 262, 2
60, 2549, 3840, 851, 1930, 851, 159094, 3062, 25029, 14612, 260, 1268, 7126, 931, 84335, 260, 24190, 70151, 260, 232910, 15420
8, 262, 260, 679, 260, 8046, 682, 11191, 94735, 653, 60796, 52623, 763, 260, 3897, 52043, 1173, 6148, 70905, 1789, 519, 98485,
653, 21804, 41603, 260, 109106, 23793, 27741, 262, 260, 14501, 2379, 260, 83067, 123553, 260, 109106, 23793, 27741, 902, 260, 6
41, 2761, 150747, 55130, 695, 30135, 34698, 13723, 25934, 1450, 260, 109106, 23793, 9133, 260, 1268, 260, 24233, 2114, 519, 90
2, 260, 1527, 260, 31902, 931, 4882, 1450, 53631, 73890, 2919, 260, 74366, 6168, 262, 5567, 16359, 260, 590, 2761, 261, 195400,
180697, 1780, 261, 2], 'token_type_ids': [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, 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], 'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}
```


Εικόνα 4.12 Αποτέλεσμα encoding κειμένου TestText2

Παρατηρούμε ότι απαιτήθηκαν 165 tokens για την κωδικοποίηση του κειμένου, ενώ αντίστοιχα αποτελέσματα (με 164 tokens) προκύπτουν και κατά το encoding του πρώτου δοκιμαστικού κειμένου (TestText1). Εκτελώντας την ίδια διαδικασία άλλες τρεις φορές κάνοντας χρήση των τριών εκτεταμένων εκδοχών του αρχικού μοντέλου, παρατηρούμε ότι ο αριθμός των απαιτούμενων tokens για την κωδικοποίηση των ίδιων κειμένων μειώνεται, το οποίο είναι γενικά επιθυμητό αφού μια συντομότερη/απλούστερη κωδικοποίηση διευκολύνει τα μοντέλα στην εξαγωγή συμπερασμάτων. Η μείωση του αριθμού των tokens γίνεται πιο αισθητή στην κωδικοποίηση του κειμένου TestText1, κάτι που είναι λογικό αφού αποτελείται ολοκληρωτικά από λέξεις των tweets που χρησιμοποιήθηκαν για τις διαδοχικές επεκτάσεις του λεξικού του μοντέλου. Ακολουθεί συγκεντρωτικός πίνακας με τον αριθμό των παραγόμενων tokens ανά δοκιμαστικό κείμενο και χρησιμοποιούμενο μοντέλο.

Πίνακας 4.1 Παραγόμενα tokens ανά κείμενο και μοντέλο

Tokenizer	***Tokens κειμένου από tweets (1)***	***Tokens κειμένου από άρθρο (2)***
mDeBERTa-v3-base-mnli-xnli	164	165
Extended1DeBERTa	154	159
Extended001DeBERTa	112	138
ExtendedAllDeBERTa	104	123

Από την άλλη, η ανασύσταση του αρχικού κειμένου (decoding) αναδεικνύει ότι η ποιότητα του tokenization, ενώ είναι άψογη κάνοντας χρήση του βασικού λεξικού, γίνεται διαδοχικά χειρότερη όσο χρησιμοποιούνται εκδοχές του μοντέλου με περισσότερες νέες πρόσθετες λέξεις - κάτι που γίνεται διαδοχικά περισσότερο αισθητό στο γενικότερο κείμενο (TestText2). Ακολουθούν εικόνες που εμφανίζουν τα αποτελέσματα του decoding του δεύτερου δοκιμαστικού κειμένου, κάνοντας χρήση αρχικού και του πλέον εκτεταμένου μοντέλου (ExtendedAllDeBERTa).

```
Out[5]: 'Το νομοσχέδιο για πρώτη φορά αναφέρεται συγκεκριμένα στους παρόχους υπηρεσιών ύδατος ως δημόσιους και δημοτικούς οργανισμούς,
δεν ασχολείται σε κανένα άρθρο του με το ιδιοκτησιακό καθεστώς, ούτε με τη μετοχική σύνθεση των παρόχων υπηρεσιών ύδατος, η νέα
Ρυθμιστική Αρχή έχει καθαρά εποπτικές και γνωμοδοτικές αρμοδιότητες, ενώ οι κανονιστικές αρμοδιότητες για την κοστολόγηση του ν
ερού παραμένουν αρμοδιότητα των υπουργών και για τον σκοπό αυτόν εκδίδουν ΚΥΑ, είπε ο κ. Καππάτος.'

--- ΑΡΧΙΚΟ ---
Το νομοσχέδιο για πρώτη φορά αναφέρεται συγκεκριμένα στους παρόχους υπηρεσιών ύδατος ως δημόσιους και δημοτικούς οργανισμούς,
δεν ασχολείται σε κανένα άρθρο του με το ιδιοκτησιακό καθεστώς, ούτε με τη μετοχική σύνθεση των παρόχων υπηρεσιών ύδατος, η νέα
Ρυθμιστική Αρχή έχει καθαρά εποπτικές και γνωμοδοτικές αρμοδιότητες, ενώ οι κανονιστικές αρμοδιότητες για την κοστολόγηση του
νερού παραμένουν αρμοδιότητα των υπουργών και για τον σκοπό αυτόν εκδίδουν ΚΥΑ, είπε ο κ. Καππάτος.
```

Εικόνα 4.13 Decoding βάσει αρχικού μοντέλου

```
Out[5]: 'Το νομοσχέδιο για πρώτη φορά αναφέρεται συγκεκριμένα στους παρόχους υπηρεσιών ύδ ατ ος ως δημόσιους και δημοτικούς οργανισμού
ς, δεν ασχολείται σε κανένα άρθρο του με το ιδιοκτησιακό καθεστώς, ούτε με τη μετ οχι κή σύνθεση των παρόχων υπηρεσιών ύδ ατ ο
ς, η νέα Ρυθμισ τικ ή Αρχή έχε ι καθαρά εποπ τικ ές και γνωμοδ οτ ικές αρμοδιότητες, ενώ οι κα νον ι στικές αρμοδιότητες για τη
ν κοστολόγηση του νερού παραμένουν αρμοδιότητα των υπουργών και για τον σκοπό αυτόν εκδί δουν ΚΥΑ, είπε ο κ. Καππάτος.'

--- ΑΡΧΙΚΟ ---
Το νομοσχέδιο για πρώτη φορά αναφέρεται συγκεκριμένα στους παρόχους υπηρεσιών ύδατος ως δημόσιους και δημοτικούς οργανισμούς,
δεν ασχολείται σε κανένα άρθρο του με το ιδιοκτησιακό καθεστώς, ούτε με τη μετοχική σύνθεση των παρόχων υπηρεσιών ύδατος, η νέα
Ρυθμιστική Αρχή έχει καθαρά εποπτικές και γνωμοδοτικές αρμοδιότητες, ενώ οι κανονιστικές αρμοδιότητες για την κοστολόγηση του
νερού παραμένουν αρμοδιότητα των υπουργών και για τον σκοπό αυτόν εκδίδουν ΚΥΑ, είπε ο κ. Καππάτος.
```

Εικόνα 4.14 Decoding βάσει εκτεταμένου μοντέλου

Κατά την επέκταση του πίνακα αναπαραστάσεων (embeddings matrix) του μοντέλου, που ακολούθησε την προσθήκη των νέων λέξεων στο λεξικό του tokenizer ώστε αυτές να μπορούν να χρησιμοποιηθούν, κάθε νέα λέξη αρχικοποιήθηκε με μία τυχαία αριθμητική αναπαράσταση (vector). Το αποτέλεσμα είναι το μοντέλο να μην έχει σωστή εικόνα της σημασίας και του τρόπου χρήσης και αλληλεπίδρασης των νέων λέξεων, όπως θα συνέβαινε αν είχε εκπαιδευτεί εξαρχής στο σύνολο του λεξικού και είχε παράξει σωστά όλα τα embedding vectors. Κάτι τέτοιο όμως απαιτεί πολλούς πόρους από πλευράς χρόνου επεξεργασίας και όγκου κειμένων για την αρχική εκπαίδευση, και ξεφεύγει από τους σκοπούς της παρούσας διπλωματικής εργασίας. Για αυτόν τον λόγο, στο τελευταίο κομμάτι της συναισθηματικής ανάλυσης θα χρησιμοποιηθεί το αρχικό εκπαιδευμένο μοντέλο χωρίς επέκταση του λεξικού του.

Όπως εξηγήθηκε στο προηγούμενο κεφάλαιο, το τελευταίο βήμα της προγραμματιστικής υλοποίησης της συναισθηματικής ανάλυσης είναι το κομμάτι του κώδικα που θα διαβάσει το προ-επεξεργασμένο dataset και, κάνοντας χρήση του επιλεγμένου μοντέλου NLI, θα παράξει στατιστικά στοιχεία για τα επικρατέστερα συναισθήματα στα tweets του dataset. Επειδή όμως

το dataset περιλαμβάνει δεκάδες χιλιάδες μη-επισημασμένα (unlabeled) μηνύματα, δεν υπάρχει κάποιος άμεσος τρόπος επιβεβαίωσης των στατιστικών αποτελεσμάτων που θα παράξει το μοντέλο.

Για να αποκτήσουμε μια εικόνα της αποτελεσματικότητας των προβλέψεων του μοντέλου, δημιουργήσαμε 4 μικρά πακέτα από tweets, με το καθένα να περιέχει μηνύματα αποκλειστικά ενός εκ των τεσσάρων βασικών συναισθημάτων που επιθυμούμε να αξιολογηθούν (1-Joy.csv, 2-Sad.csv, 3-Anger.csv, 4-Fear.csv). Τα μηνύματα που περιλαμβάνονται στα πακέτα αυτά είναι πολιτικής αλλά και γενικότερης θεματολογίας, παρουσιάζοντας το εκάστοτε συναίσθημα σχετικά ξεκάθαρα αλλά σε διαφορετικούς βαθμούς. Στην επόμενη εικόνα παρουσιάζεται ενδεικτικά το περιεχόμενο του πακέτου μηνυμάτων 1-Joy.csv, το οποίο περιέχει tweets με κυρίαρχο συναίσθημα την “χαρά”.

Index	Final
0	Ευτυχώς στο τελευταίο 6μηνο δείχνει μια αλλαγή προς το καλύτερο, ειδικά από νεότερα στην ηλικία στελέχη.
1	Με τιμά ιδιαίτερα η εμπιστοσύνη που μου δείχνουν οι συντρόφισσες και σύντροφοι της νεολαίας ΣΥΡΙΖΑ.
2	Ξαδέρφη να είναι ευτυχισμένο το παιδάκι σου, να το χαίρεσαι.
3	Καθε φορά που ένα στερεότυπο καταρρεει ο κόσμος γίνεται λίγο καλυτερος.
4	Καλή η εικόνα των ψηφοδελτίων τους. Έχουν πολλούς νέους με καλή επαγγελματική και κοινωνική πορεία και πολλούς καταξιωμένους με ευρύτερη παιδεία και ειδικές γνώσεις.
5	Χαίρομαι να βλέπω αθλητές μας να σηκώνουν τη Γαλανόλευκη ψηλά.
6	Πάντα μέσα στην οικογένεια μου έβρισκα την αγάπη, την στήριξη, και την γαλήνη της ψυχής μου.
7	Ο έρωτας είναι φωτισμένα πρόσωπα, είναι χαμόγελα, είναι το πρόσωπο της ευτυχίας και μπορεί να είναι δημόσιος.
8	Ο έρωτας είναι συναίσθημα μοναδικό και είναι τυχεροί όσοι το έχουν ζήσει αμοιβαία.
9	Είμαι στην ευχάριστη θέση να σας ενημερώσω πως δημοσιεύτηκε η 1η δουλειά μας, σε Αμερικάνικο ΜΜΕ.
10	Δεν θυμάμαι να ήμουν ποτέ ξανά τόσο ευτυχισμένος και περήφανος με την ομάδα μου. Υγεία να έχετε ρε μάγκες.
11	Δικαιωμένη η εφημερίδα μας για άλλη μια φορά.

Εικόνα 4.15 Πακέτο μηνυμάτων χαράς

Εκτελώντας τον κώδικα ανάλυσης συναισθήματος για κάθε ένα από τα 4 πακέτα προαξιολογημένων μηνυμάτων, επιστρέφονται τα στατιστικά που φαίνονται στις παρακάτω εικόνες:

```
== Αποτελέσματα ανάλυσης ( 1-Joy.csv ) ==

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0   χαρά   100.0

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0    χαρά    91.7
1   φόβος     8.3

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0      χαρά    91.7
1     φόβος     8.3
```

Εικόνα 4.16 Ανάλυση μηνυμάτων χαράς

```
== Αποτελέσματα ανάλυσης ( 2-Sad.csv ) ==

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0   λύπη    80.0
1  θυμός     6.7
2  φόβος     6.7
3   χαρά     6.7

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0    λύπη    46.7
1   θυμός    33.3
2   φόβος    13.3
3    χαρά     6.7

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0      λύπη    73.3
1     θυμός    13.3
2     φόβος     6.7
3      χαρά     6.7
```

Εικόνα 4.17 Ανάλυση μηνυμάτων λύπης

```
== Αποτελέσματα ανάλυσης ( 3-Anger.csv ) ==

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0  θυμός    50.0
1   λύπη    28.6
2  φόβος    14.3
3   χαρά     7.1

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0   θυμός    64.3
1   φόβος    28.6
2    λύπη     7.1

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0     θυμός    64.3
1      λύπη    21.4
2     φόβος    14.3
```

Εικόνα 4.18 Ανάλυση μηνυμάτων θυμού

```
== Αποτελέσματα ανάλυσης ( 4-Fear.csv ) ==

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0  φόβος    84.6
1   λύπη    15.4

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0   φόβος    92.3
1    λύπη     7.7

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0     φόβος    84.6
1      λύπη    15.4
```

Εικόνα 4.19 Ανάλυση μηνυμάτων φόβου

Παρατηρούμε ότι και στις τέσσερις περιπτώσεις, το κυρίαρχο συναίσθημα αναγνωρίστηκε σωστά και από τις τρεις διαθέσιμες μεθόδους αξιολόγησης, με διαφορετικά ποσοστά επιτυχίας. Για τρία από τα συναισθήματα (χαρά, λύπη, φόβος) το μοντέλο απάντησε με πολύ υψηλά ποσοστά συμφωνίας (100%, 80% και 84,6% αντίστοιχα) με την προαξιολόγηση που είχε

γίνει κατά την δημιουργία των μικρών αυτών συνόλων - σε αυτές τις περιπτώσεις, όπου το μοντέλο επέδειξε μεγάλη “αυτοπεποίθηση”, η μέθοδος αξιολόγησης βάσει μικρότερου δείκτη αβεβαιότητας (BestGuess) δεν επέφερε ουσιαστικές αλλαγές στα αποτελέσματα. Αντίθετα, στο τέταρτο συναίσθημα (θυμός), όπου το ποσοστό επιτυχίας του μοντέλου βάσει της μέγιστης συμφωνίας (MaxEnt) ήταν μικρότερο σε σχέση με τα υπόλοιπα (50%), η μέθοδος BestGuess βελτίωσε αρκετά την πρόβλεψη ανεβάζοντας το ποσοστό επιτυχίας σε 64,3%. Τα συγκεντρωτικά ποσοστά σωστών προβλέψεων στα μικρά αυτά σύνολα, εμφανίζονται στον παρακάτω πίνακα.

Πίνακας 4.2 Επιτυχείς προβλέψεις ανά μέθοδο και συναίσθημα

Συναίσθημα	**Μέθοδος MaxEnt (%)**	**Μέθοδος BestGuess (%)**
Χαρά	100	91,7
Λύπη	80	73,3
Θυμός	50	64,3
Φόβος	84,6	84,6
Μέσος όρος =>	78,7	78,5

Υπολογίζοντας τον μέσο όρο απόδοσης των δύο μεθόδων στα 4 συναισθήματα, παρατηρούμε ότι η συνολική απόδοσή τους είναι πρακτικά ίδια. Η BestGuess όμως, συνυπολογίζοντας τα αποτελέσματα της MinCont ειδικά σε συγκεκριμένες περιπτώσεις όπου το μοντέλο δεν επιδεικνύει μεγάλη “αυτοπεποίθηση”, δυνητικά μπορεί να δώσει πιο εξισορροπημένα αποτελέσματα μεταξύ των διαφορετικών συναισθημάτων. Σε κάθε περίπτωση, τα παραπάνω συγκεντρωτικά ποσοστά προσεγγίζουν σε ικανοποιητικό βαθμό την επίσημη αξιολογημένη αποτελεσματικότητα του μοντέλου mDeBERTa-v3-base-mnli-xnli, το οποίο για χρήση σε ελληνικά κείμενα ανέρχεται σε 82,6% (μέσος όρος για τις 15 υποστηριζόμενες γλώσσες 80,8%) (Laurer et al, 2022).

Δεδομένου ότι, βάσει και των παραπάνω μετρήσεων, το χρησιμοποιούμενο μοντέλο μπορεί να αναγνωρίσει τα 4 προεπιλεγμένα βασικά συναισθήματα σε ελληνικά κείμενα με ένα υψηλό ποσοστό αξιοπιστίας, το τελευταίο βήμα ήταν να εκτελέσουμε τον κώδικα συναισθηματικής ανάλυσης για τα datasets που δημιουργήσαμε αρχικά. Στις εικόνες που ακολουθούν, εμφανίζονται τα αποτελέσματα της ανάλυσής των πρώτων 20 χιλιάδων εγγραφών του καθενός από τα 2 εκτενέστερα datasets, μαζί με τους χρόνους εκτέλεσης των υπολογισμών κάνοντας χρήση 1280 πυρήνων CUDA μη-επαγγελματικής κάρτας γραφικών του 2016, μέσης απόδοσης, με 6GB μνήμης.

```
== Αποτελέσματα ανάλυσης ( PoliticalTweets-Cleaned-Part.csv )

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0  φόβος    35.5
1  θυμός    30.5
2   χαρά    17.3
3   λύπη    16.6

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0   φόβος    47.9
1   θυμός    28.9
2    λύπη    11.9
3    χαρά    11.4

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0     φόβος    45.8
1     θυμός    29.4
2      χαρά    12.8
3      λύπη    11.9


CPU times: total: 51min 20s
Wall time: 52min 50s
```

Εικόνα 4.20 Ανάλυση πολιτικών μηνυμάτων

```
== Αποτελέσματα ανάλυσης ( TempiTweets-Cleaned-Part.csv )

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0  φόβος    32.1
1  θυμός    32.0
2   λύπη    27.9
3   χαρά     8.0

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0   θυμός    36.3
1   φόβος    35.4
2    λύπη    23.5
3    χαρά     4.8

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0     φόβος    35.7
1     θυμός    34.3
2      λύπη    24.8
3      χαρά     5.2


CPU times: total: 51min 11s
Wall time: 52min 31s
```

Εικόνα 4.21 Ανάλυση μηνυμάτων Τεμπών

Παρατηρούμε ότι στα μηνύματα πολιτικής θεματολογίας κυριαρχούν τα συναισθήματα του φόβου και του θυμού, ενώ στα μηνύματα γύρω από τα Τέμπη έχει αναμενόμενα έντονη παρουσία και το συναίσθημα της λύπης. Σε αυτό το σημείο, πρέπει να επισημανθεί ότι τα αποτελέσματα συνδιαμορφώνονται από το γεγονός ότι η χρησιμοποιούμενη μέθοδος υποχρεώνει το μοντέλο να κατηγοριοποιήσει κάθε εγγραφή των datasets σε ένα από τα προαποφασισμένα premises/συναισθήματα, παρόλο που κάποια μηνύματα αντικειμενικά μπορεί να μην αντιπροσωπεύονται ικανοποιητικά από κανένα από αυτά τα premises.

Αυτό γίνεται πιο ορατό κατά την συναισθηματική ανάλυση του τρίτου αρχικού dataset, με μηνύματα που χρησιμοποιούν θεματική επισήμανση (hashtag) ψυχαγωγικού τηλεοπτικού προγράμματος. Όπως φαίνεται στην πρώτη από τις δύο εικόνες που ακολουθούν, η “χαρά” αναγνωρίζεται ως στατιστικά πρωτεύον συναίσθημα, κάτι αναμενόμενο δεδομένου του χαρακτήρα της θεματικής επισήμανσης. Παρόλα αυτά, παρατηρείται ένα σημαντικό ποσοστό καταγραφής “φοβου”, κάτι που πιθανόν οφείλεται και στο φαινόμενο της χρήση των δημοφιλών hashtags για σκοπούς προβολής μηνυμάτων από ιστότοπους/χρήστες που δεν έχουν σχέση με την θεματική. Τέτοια μηνύματα συνήθως είναι ειδησεογραφικού χαρακτήρα, και μπορεί να μην κατηγοριοποιηθούν σωστά βάσει των διαθέσιμων συναισθημάτων/premises. Αν δοκιμάσουμε να επανεκτελέσουμε την ανάλυση προσθέτοντας μια πέμπτη επιλογή κατηγοριοποίησης χωρίς συναισθηματικό περιεχόμενο (“ουδετερότητα”), παρατηρείται η μείωση των ποσοστών των υπολοίπων επιλογών κατά ένα αθροιστικό ποσοστό 10-15%, καθώς και η συγκριτική ενίσχυση

της “χαράς” έναντι του “φόβου” βάσει όλων των τεχνικών υπολογισμού - κάτι που υποδηλώνει την ωφέλεια ύπαρξης τουλάχιστον μιας τέτοιας ουδέτερης επιλογής.

```
== Αποτελέσματα ανάλυσης ( SpitiMegaTweets

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

  MaxEnt  Counts
0   χαρά    40.2
1  φόβος    36.3
2  θυμός    14.9
3   λύπη     8.6

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

  MinCont  Counts
0   φόβος    40.3
1    χαρά    38.7
2   θυμός    13.0
3    λύπη     8.0

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

  BestGuess  Counts
0      χαρά    40.8
1     φόβος    37.3
2     θυμός    13.9
3      λύπη     8.0
```

Εικόνα 4.22 Ανάλυση χωρίς ουδέτερο premise

```
== Αποτελέσματα ανάλυσης ( SpitiMegaTweets

Ποσόστωση tweets ανά συναίσθημα
βάσει max entailment (%) =>

         MaxEnt  Counts
0          χαρά    37.1
1         φόβος    31.2
2  ουδετερότητα    15.0
3         θυμός    10.9
4          λύπη     5.8

Ποσόστωση tweets ανά συναίσθημα
βάσει min contradiction (%) =>

        MinCont  Counts
0         φόβος    38.4
1          χαρά    37.9
2         θυμός    12.2
3          λύπη     6.4
4  ουδετερότητα     5.1

Ποσόστωση tweets ανά συναίσθημα
βάσει min δείκτη αβεβαιότητας (%) =>

      BestGuess  Counts
0          χαρά    39.5
1         φόβος    33.2
2         θυμός    11.1
3  ουδετερότητα    10.7
4          λύπη     5.5
```

Εικόνα 4.23 Ανάλυση με ουδέτερο premise

5. Συμπεράσματα και προτάσεις

Όπως εξηγήθηκε στην εισαγωγή, η παρούσα διπλωματική εργασία επεδίωξε να κάνει μια διερεύνηση της εφικτότητας και αξιολόγηση της αποτελεσματικότητας μιας εναλλακτικής προσέγγισης στην ανάλυση συναισθήματος, αντί της διαδεδομένης χρήσης προαξιολογημένων συναισθηματικών λεξικών. Η προσέγγιση αυτή αφορά την χρήση μοντέλων μηχανικής μάθησης προεκπαιδευμένων σε εργασίες NLI (Natural Language Inference - Συμπερασμάτων Φυσικής Γλώσσας) για την ανάλυση συναισθήματος μέσω τεχνικής zero-shot classification (ταξινόμησης μηδενικής βολής), δηλαδή χωρίς το μοντέλο να έχει “δει” επιτυχημένα παραδείγματα της εργασίας που του ζητείται να επιτελέσει. Ακόμα πιο ειδικά, η προαναφερθείσα διαδικασία ανάλυσης συναισθήματος εφαρμόστηκε σε ελληνικά κείμενα - στην περίπτωσή μας σε μηνύματα του κοινωνικού δικτύου Twitter, οπότε το πρώτο μέρος της διπλωματικής εργασίας ασχολήθηκε με την παρουσίαση ενός τμήματος του API (Application Programming Interface -

Προγραμματιστική Διεπαφή) του κοινωνικού δικτύου. Τελικός στόχος ήταν τα εργαλεία αυτά να μπορούν να αξιοποιηθούν ως μέσο για μια γρήγορη αναγνώριση του δημοσίου συναισθήματος γύρω από ένα θέμα, πρόσωπο ή γεγονός.

Τα πειραματικά αποτελέσματα, από την συλλογή και την βελτίωση των datasets έως την αναγνώριση των βασικών συναισθημάτων από το επιλεγμένο μοντέλο, θα μπορούσαν να χαρακτηριστούν ενθαρρυντικά. Παρά το γεγονός ότι έως και την ολοκλήρωση της συγγραφής της εργασίας δεν είχε εγκριθεί από το Twitter το αίτημα παροχής Ακαδημαϊκής πρόσβασης στο API, η πρόσβαση Βασικού επιπέδου που είναι διαθέσιμη χωρίς προϋποθέσεις σε όλους (έως και αυτή την ώρα) αποδεικνύεται αρκετή για την δημιουργία datasets ικανού αριθμού μηνυμάτων των τελευταίων 7 ημερών για σκοπούς διερεύνησης της κοινής γνώμης. Η απόπειρα χρησιμοποίησης των datasets αυτών για τον εμπλουτισμό και την περαιτέρω εξειδίκευση του λεξικού του μοντέλου μηχανικής μάθησης, φάνηκε ότι δεν είναι από μόνη της αρκετή, όταν δεν συνοδεύεται από επανεκπαίδευση του μοντέλου ώστε να αποτυπωθούν σωστά οι έννοιες και οι αλληλεπιδράσεις όλων των λέξεων μέσω των embedding vectors (διανυσματικών αναπαραστάσεων). Παρόλα αυτά, το προεκπαιδευμένο πολυγλωσσικό μοντέλο συμπερασμάτων mDeBERTa-v3-base-mnli-xnli, που επιλέχθηκε ως βασισμένο σε ένα από τα πιο σύγχρονα και ικανά μοντέλα τύπου transformer (DeBERTa), επέδειξε πολύ καλές δυνατότητες προσαρμογής στην εργασία αναγνώρισης συναισθημάτων μέσα σε ελληνικά κείμενα, προσεγγίζοντας την μέγιστη αξιολογημένη απόδοσή του σε γενικές εργασίες NLI (81-83%), όταν χρειάστηκε να διακρίνει μεταξύ 4 βασικών συναισθημάτων (κάτι που πιθανόν είναι αρκετό για μια γρήγορα ανάλυση για σκοπούς marketing).

Ως τελικό συμπέρασμα, θα μπορούσε να αναφερθεί ότι η χρήση μοντέλων NLI και τεχνικών zero-shot classification, παρόλο που ακόμα δεν επιτυγχάνει την αποτελεσματικότητα των καλύτερων supervised μοντέλων κατηγοριοποίησης με χρήση λεξικού (μπορεί να φτάσει ακόμα και σε ποσοστά άνω του 92% όσο μειώνεται ο αριθμός των κλάσεων και αναλόγως της γλώσσας) (Duwairi et al, 2015), είναι μια καλή λύση που μπορεί να δώσει αποτελέσματα γρήγορα και εύκολα, όταν δεν διαθέτουμε προαξιολογημένα (labeled) δεδομένα αλλά υπάρχει αφθονία πρωτογενών κειμένων (όπως συμβαίνει με τα κοινωνικά δίκτυα).

Δεδομένου ότι, όπως εξηγήθηκε αρχικά, σκοπός της εργασίας δεν ήταν η δημιουργία μιας άρτιας και ολοκληρωμένης λύσης, αλλά η διερεύνηση εφικτότητας και μια πρώτη αξιολόγηση των δυνατοτήτων της επιλεγμένης προσέγγισης, υπάρχει πληθώρα πεδίων που προσφέρονται για μελλοντική έρευνα και βελτιώσεις. Ενδεικτικά αναφέρονται:

- η αξιοποίηση μηνυμάτων από χρονικά διαστήματα παλαιότερα των τελευταίων 7 ημερών, μέσω ανώτερων επιπέδων πρόσβασης στο Twitter API, τόσο για ιστορικούς λόγους, όσο και για την δημιουργία πιο αναλυτικών λεξικών για τα μοντέλα.
- ο πειραματισμός με τα διαθέσιμα στάδια επεξεργασίας των πρωτογενών datasets, αλλά και με άλλα που τυχόν δεν περιλαμβάνονται/υλοποιήθηκαν στην εργασία αυτή. Για παράδειγμα, την αντικατάσταση των emoticons/emojis που χρησιμοποιούνται ευρέως στα κοινωνικά δίκτυα με αντιπροσωπευτικό κείμενο που θα φέρει συναισθηματικό περιεχόμενο, την κατάλληλη επεξεργασία των λέξεων με απόστροφο, την αντιμετώπιση λανθασμένης αναγνώρισης άγνωστων όρων ως ορθογραφικών λαθών, την αξιοποίηση τρίτων βιβλιοθηκών για σωστή λημματοποίηση της ελληνικής γλώσσας και άλλα.
- η επανεκπαίδευση των συνδυασμών μοντέλων-tokenizers πάνω σε μεγάλα datasets ή το fine-tuning πάνω σε εξειδικευμένα σύνολα από tweets, για την βελτίωση της αποτελεσματικότητας των μοντέλων στις εργασίες συναισθηματικής ανάλυσης σε τέτοια κείμενα.
- η διερεύνηση της αποτελεσματικότητας και της διακριτικής ικανότητας των μοντέλων όταν περιλαμβάνονται και υποθέσεις χωρίς συναισθηματικό περιεχόμενο (για παράδειγμα “ουδετερότητα” ή “αδιαφορία”). Επίσης, περιλαμβάνοντας περισσότερα των 4 βασικών συναισθημάτων που επιλέχθηκαν για τις ανάγκες αυτής της εργασίας, ειδικά όταν αυτά τα συναισθήματα διαθέτουν αλληλοεπικαλυπτόμενα χαρακτηριστικά (όπως για παράδειγμα χαρά, αγάπη, έκπληξη).
- ο πειραματισμός ως προς την απόδοση των μοντέλων NLI με εναλλακτικές διατυπώσεις των υποθέσεων (για παράδειγμα “Νιώθει χαρά” ή “Το συναίσθημα είναι χαρά” αντί “χαρά”), καθώς τα αποτελεσματά τους επηρεάζονται από αυτές τις αλλαγές.
- η απόπειρα εφαρμογής της παραπάνω μεθόδου σε κείμενα που συλλέγονται σε πραγματικό χρόνο, μέσω του Filtered stream endpoint
- η χρησιμοποίηση τυχόν νεότερων και πιο αποδοτικών (πολυγλωσσικών ή ακόμα και ελληνικών) μοντέλων, εκπαιδευμένων σε εργασίες NLI.

Λίστα βιβλιογραφικών αναφορών

Aggarwal, C.C., 2015. Data mining: the textbook (Vol. 1). New York: springer.

Aparna, U.R. and Paul, S., 2016, November. Feature selection and extraction in data mining. In 2016 Online international conference on green engineering and technologies (IC-GET) (pp. 1-3). IEEE.

Auxier, B. and Anderson, M., 2021. Social media use in 2021. Pew Research Center, 1, pp.1-4.

Bhaskar, J., Sruthi, K. and Nedungadi, P., 2015. Hybrid approach for emotion classification of audio conversation based on text and speech mining. Procedia Computer Science, 46, pp.635-643.

Bowman, S.R., Angeli, G., Potts, C. and Manning, C.D., 2015. A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S., 2020. Language models are few-shot learners. Advances in neural information processing systems, 33, pp.1877-1901.

Burchfiel, A., 2022. What is NLP (Natural Language Processing) Tokenization? [Online]. Available at: https://www.tokenex.com/blog/ab-what-is-nlp-natural-language-processing-tokenization/ (Accessed: 14 March 2023).

Cardie, C., 2014. Sentiment Analysis and Opinion Mining Bing Liu (University of Illinois at Chicago) Morgan & Claypool (Synthesis Lectures on Human Language Technologies, edited by Graeme Hirst, 5 (1)), 2012, 167 pp; paperbound, ISBN 978-1-60845-884-4.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y., 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078.

Conneau, A., Lample, G., Rinott, R., Williams, A., Bowman, S.R., Schwenk, H. and Stoyanov, V., 2018. XNLI: Evaluating cross-lingual sentence representations. arXiv preprint arXiv:1809.05053.

Couto, J., 2015. The Definitive Guide to Natural Language Processing (NLP) [Online]. Available at: https://monkeylearn.com/blog/definitive-guide-natural-language-processing/ (Accessed: 14 March 2023).

Darwich, M., Mohd, S.A., Omar, N. and Osman, N.A., 2019. Corpus-Based Techniques for Sentiment Lexicon Generation: A Review. J. Digit. Inf. Manag., 17(5), p.296.

Delua, J., 2021. Supervised vs. Unsupervised Learning: What's the Difference? [Online]. Available at: https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning/ (Accessed: 14 March 2023).

Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Duwairi, R.M., Ahmed, N.A. and Al-Rifai, S.Y., 2015. Detecting sentiment embedded in Arabic social media–a lexicon-based approach. Journal of Intelligent & Fuzzy Systems, 29(1), pp.107-117.

Ekman, P., 1992. An argument for basic emotions. Cognition & emotion, 6(3-4), pp.169-200.

Fang, Z., Dudek, J. and Costas, R., 2020. The stability of Twitter metrics: A study on unavailable Twitter mentions of scientific publications. Journal of the Association for Information Science and Technology, 71(12), pp.1455-1469.

Forvis, 2020. Natural Language Processing for Sentiment Analysis [Online]. Available at: https://www.forvis.com/article/natural-language-processing-for-sentiment-analysis (Accessed: 14 March 2023).

Gaisbauer, F., Pournaki, A., Banisch, S. and Olbrich, E., 2021. Ideological differences in engagement in public debate on Twitter. Plos one, 16(3), p.e0249241.

Gillioz, A., Casas, J., Mugellini, E. and Abou Khaled, O., 2020, September. Overview of the Transformer-based Models for NLP Tasks. In 2020 15th Conference on Computer Science and Information Systems (FedCSIS) (pp. 179-183). IEEE.

Guillou, P., 2021. NLP | How to add a domain-specific vocabulary (new tokens) to a subword tokenizer already trained like BERT WordPiece [Online]. Available at: https://medium.com/@pierre_guillou/nlp-how-to-add-a-domain-specific-vocabulary-new-tokens-to-a-subword-tokenizer-already-trained-33ab15613a41/ (Accessed: 20 March 2023).

Gupta, M.K. and Chandra, P., 2020. A comprehensive survey of data mining. International Journal of Information Technology, 12(4), pp.1243-1257.

Hamzah, M. and Vu, T.T., 2018, November. A Taxonomy of Twitter Data Analytics Techniques. In Proceedings of the 32nd IBIMA Conference, Seville, Spain (pp. 15-16).

Hänska, M. and Bauchowitz, S., 2019. Can social media facilitate a European public sphere? Transnational communication and the Europeanization of Twitter during the Eurozone crisis. Social media+ society, 5(3), p.2056305119854686.

He, P., Liu, X., Gao, J. and Chen, W., 2020. Deberta: Decoding-enhanced bert with disentangled attention. arXiv preprint arXiv:2006.03654.

Hugging Face, 2022. Zero-Shot Classification [Online]. Available at: https://huggingface.co/tasks/zero-shot-classification/ (Accessed: 14 March 2023).

Jungherr, A., 2014. Twitter in politics: a comprehensive literature review. Available at SSRN 2865150.

Kaplan, A.M. and Haenlein, M., 2010. Users of the world, unite! The challenges and opportunities of Social Media. Business horizons, 53(1), pp.59-68.

Lample, G. and Conneau, A., 2019. Cross-lingual language model pretraining. arXiv preprint arXiv:1901.07291.

Laurer, M., Atteveldt, W.V., Casas, A. and Welbers, K., 2022. Less annotating, more classifying–addressing the data scarcity issue of supervised machine learning with deep transfer learning and bert-nli.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L. and Stoyanov, V., 2019. Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692.

Maryville University, 2020. The Evolution of Social Media: How Did It Begin, and Where Could It Go Next? [Online]. Available at: https://online.maryville.edu/blog/evolution-social-media/ (Accessed: 14 March 2023).

Medhat, W., Hassan, A. and Korashy, H., 2014. Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4), pp.1093-1113.

Mislove, A., Lehmann, S., Ahn, Y.Y., Onnela, J.P. and Rosenquist, J., 2011. Understanding the demographics of Twitter users. In Proceedings of the International AAAI Conference on Web and Social Media (Vol. 5, No. 1, pp. 554-557).

Nandwani, P. and Verma, R., 2021. A review on sentiment analysis and emotion detection from text. Social Network Analysis and Mining, 11(1), p.81.

Pai, A., 2020. What is Tokenization in NLP? Here's All You Need To Know [Online]. Available at: https://www.analyticsvidhya.com/blog/2020/05/what-is-tokenization-nlp/ (Accessed: 14 March 2023).

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D. and Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI blog, 1(8), p.9.

Ramteke, J., Shah, S., Godhia, D. and Shaikh, A., 2016, August. Election result prediction using Twitter sentiment analysis. In 2016 international conference on inventive computation technologies (ICICT) (Vol. 1, pp. 1-5). IEEE.

Romero, C. and Ventura, S., 2013. Data mining in education. Wiley Interdisciplinary Reviews: Data mining and knowledge discovery, 3(1), pp.12-27.

Rosen, A. and Ihara, I., 2017. Giving you more characters to express yourself [Online]. Available at: https://blog.twitter.com/en_us/topics/product/2017/Giving-you-more-characters-to-express-yourself/ (Accessed: 14 March 2023).

Sanh, V., Debut, L., Chaumond, J. and Wolf, T., 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.

Shibata, Y., Kida, T., Fukamachi, S., Takeda, M., Shinohara, A., Shinohara, T. and Arikawa, S., 1999. Byte Pair encoding: A text compression scheme that accelerates pattern matching.

Shree, P., 2020. The Journey of Open AI GPT models [Online]. Available at: https://medium.com/walmartglobaltech/the-journey-of-open-ai-gpt-models-32d95b7b7fb2/ (Accessed: 14 March 2023).

Siqueira, H. and Barros, F., 2010, October. A feature extraction process for sentiment analysis of opinions on services. In Proceedings of International Workshop on Web and Text Intelligence (pp. 404-413).

Srivastava, A., 2022. What Are Transformers In NLP And Its Advantages [Online]. Available at: https://blog.knoldus.com/what-are-transformers-in-nlp-and-its-advantages/ (Accessed: 14 March 2023).

Sun, S., Luo, C. and Chen, J., 2017. A review of natural language processing techniques for opinion mining systems. Information fusion, 36, pp.10-25.

Symeonidis, S., Effrosynidis, D. and Arampatzis, A., 2018. A comparative evaluation of pre-processing techniques and their interactions for twitter sentiment analysis. Expert Systems with Applications, 110, pp.298-310.

Tai, W., Kung, H.T., Dong, X.L., Comiter, M. and Kuo, C.F., 2020, November. exBERT: Extending pre-trained models with domain-specific vocabulary under constrained training resources. In Findings of the Association for Computational Linguistics: EMNLP 2020 (pp. 1433-1439).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems, 30.

Williams, A., Nangia, N. and Bowman, S.R., 2017. A broad-coverage challenge corpus for sentence understanding through inference. arXiv preprint arXiv:1704.05426.

Wójcik, R., 2019. Unsupervised Sentiment Analysis [Online]. Available at: https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483/ (Accessed: 14 March 2023).

Yang, A., Zhang, W., Wang, J., Yang, K., Han, Y. and Zhang, L., 2020. Review on the application of machine learning algorithms in the sequence data mining of DNA. Frontiers in Bioengineering and Biotechnology, 8, p.1032.

Παράρτημα Α: Κώδικας υλοποίησης σε Python

Α.1 Collection

```
#Για την αποστολή των requests
import requests

#Για την αποθήκευση των κωδικών πρόσβασης
import os

#Για διαχείριση JSON λιστών (φορμάτ απάντησης του Twitter API)
import json

#Για διαχείριση .csv αρχείων
import csv

#Για μετατροπή των ημερομηνιών που λαμβάνουμε από το API
import datetime
import dateutil.parser
import unicodedata

#Για προσθήκη αναμονών μεταξύ των requests
import time
```

```
#Αποθήκευση του access token σε environmental variable (συστήνεται το σβήσιμο μετά την εκτέλεση)

os.environ['TOKEN'] = '##################################################'
```

```
#Συνάρτηση ανάκτησης του access token

def auth():
    return os.getenv('TOKEN')
```

```
#Συνάρτηση δημιουργίας του authentication string

def create_headers(bearer_token):
    headers = {"Authorization": "Bearer {}".format(bearer_token)}
    return headers
```

```
#Συνάρτηση επιστροφής παραμέτρων αναζήτησης

def create_url(keyword, start_date, end_date, max_results = 10):

    #Από εδώ διαλέγουμε ποια έκδοση του API θα χρησιμοποιήσουμε (αλλαγή για Academic)
    search_url = "https://api.twitter.com/2/tweets/search/recent"

    #Διαθέσιμοι παράμετροι αναζήτησης - δεν είναι όλοι απαραίτητοι.
    #Παρακάτω ένα αντίγραφο με αυτούς που χρειαζόμαστε.

    #query_params = {'query': keyword,
    #                'start_time': start_date,
    #                'end_time': end_date,
    #                'max_results': max_results,
    #                'expansions': 'author_id,in_reply_to_user_id,geo.place_id',
    #                'tweet.fields':
'id,text,author_id,in_reply_to_user_id,geo,conversation_id,created_at,lang,public_metrics,referenced_tweets,r
```

```
eply_settings,source',
    #             'user.fields': 'id,name,username,created_at,public_metrics,verified',
    #             'place.fields': 'full_name,id,country,country_code,geo,name,place_type',
    #             'next_token': {}}

    query_params = {'query': keyword,
                    'start_time': start_date,
                    'end_time': end_date,
                    'max_results': max_results,
                    'tweet.fields': 'id,text,created_at,author_id',
                    'next_token': {}}

    return (search_url, query_params)
```

```
#Συνάρτηση σύνδεσης στο API (το 200 είναι ο κωδικός επιτυχούς σύνδεσης)

def connect_to_endpoint(url, headers, params, next_token = None):
    params['next_token'] = next_token   #Αν υπάρχει, όπως προκύπτει από επαναλαμβανόμενες κλήσεις της create_url
    response = requests.request("GET", url, headers = headers, params = params)
    print("*** Κωδικός απάντησης API: " + str(response.status_code))
    if response.status_code != 200:
        raise Exception(response.status_code, response.text)
    return response.json()
```

```
#Παράμετροι του αιτήματος προς το API (χωρίς Academic access, υπάρχει περιορισμός αποτελεσμάτων έως 1 εβδομάδας πριν το αίτημα και έως 100 tweets ανά "σελίδα" απαντήσεων

bearer_token = auth()
headers = create_headers(bearer_token)

#keyword = "(Μητσοτακη OR Τσιπρα OR ΝΔ OR συριζα OR εκλογ OR βουλη OR βουλευτ OR κιναλ OR ανδρουλακη OR κκε OR κουτσουμπα OR βελοπουλ OR βαρουφακη OR μερα25) lang:el -is:retweet"
keyword = "(Τεμπη OR Λαρισα OR τρενο) lang:el -is:retweet"

start_time_list = ['2023-02-28T10:00:00.000Z']
end_time_list =   ['2023-03-07T06:00:00.000Z']  #-3 ώρες από Ελλάδα
max_results = 100

#Αριθμός των tweets που λαμβάνουμε ως απάντηση, θα αυξάνεται σε κάθε loop
total_tweets = 0
```

```
#Δημιουργία κενού αρχείου αποθήκευσης των tweets της απάντησης

dataset="TweetCorpus.csv"

csvFile = open(dataset, "a", newline="", encoding='utf-8')
csvWriter = csv.writer(csvFile, delimiter='|')

#Δημιουργία τίτλων των στηλών, βάσει των πληροφοριών που ζητήσαμε ανά tweet
csvWriter.writerow(['AuthorID','TweetID','CreatedAt','TweetBody'])
csvFile.close()
```

```
#Συνάρτηση εγγραφής των πληροφοριών του κάθε tweet της "σελίδας" απαντήσεων στις ανάλογες στήλες (απενεργοποιημένα όσα δεν χρειαζόμαστε)

def append_to_csv(json_response, fileName):
```

```
    #Μετρητής
    counter = 0

    #Άνοιγμα του αρχείου απαντήσεων
    csvFile = open(fileName, "a", newline="", encoding='utf-8')
    csvWriter = csv.writer(csvFile, delimiter='|')

    #Για κάθε tweet της σελίδας
    for tweet in json_response['data']:

        #Συγγραφέας του tweet
        author_id = tweet['author_id']

        #Χρόνος δημιουργίας
        created_at = dateutil.parser.parse(tweet['created_at'])

        #Τοποθεσία (δεν αναγνωρίζεται πάντα)
        #if ('geo' in tweet):
        #    geo = tweet['geo']['place_id']
        #else:
        #    geo = " "

        #ID του tweet
        tweet_id = tweet['id']

        #Γλώσσα
        #lang = tweet['lang']

        #Διάφορα δημόσια στατιστικά
        #retweet_count = tweet['public_metrics']['retweet_count']
        #reply_count = tweet['public_metrics']['reply_count']
        #like_count = tweet['public_metrics']['like_count']
        #quote_count = tweet['public_metrics']['quote_count']

        #Πηγή (πχ όνομα εφαρμογής που χρησιμοποιήθηκε)
        #source = tweet['source']

        #Κυρίως κείμενο του tweet (αντικαθιστούμε on-the-fly τις τυχόν αλλαγές γραμμής με κενό
        #ώστε κάθε tweet να αποθηκεύεται σωστά στο .csv, σε μία γραμμή)
        text = tweet['text']
        text = text.replace('|',")
        noenterstext = text.replace('\n',")

        #Μορφοποίηση των πληροφοριών ως λίστα, για εγγραφή στο .csv
        res = [author_id, tweet_id, created_at, noenterstext]

        #Εγγραφή του tweet σε νέα γραμμή
        csvWriter.writerow(res)
        counter += 1

    #Κλείσιμο του .csv όταν αποθηκευτούν όλα τα tweets της σελίδας
    csvFile.close()
```

```
#Βρόχος συλλογής των tweets

#Για κάθε ένα από τα (τυχόν πολλαπλά) χρονικά σημεία έναρξης που δηλώσαμε παραπάνω
```

```
for i in range(0,len(start_time_list)):

    print("-------------------------------------")
    print("\n")

    #Παράμετροι εισόδου
    next_token = None
    count = 0 #Μετρητής tweets ανά χρονικό σημείο έναρξης
    max_count = 100 #Αριθμός tweets ανά σελίδα (100 χωρίς academic)
    flag = True

    #Για να μπορούμε να τερματίζουμε το loop όταν τελειώνουν οι απαντήσεις
    while flag:
        if count >= max_count:
            break

        #Αρχικοποίηση όλων των απαραίτητων στοιχείων με χρήση
        #των συναρτήσεων που δημιουργήσαμε παραπάνω και των
        #παραμέτρων που δηλώσαμε:
        url = create_url(keyword, start_time_list[i],end_time_list[i], max_results)
        json_response = connect_to_endpoint(url[0], headers, url[1], next_token)
        result_count = json_response['meta']['result_count']

        #Αν η απάντηση περιέχει και "next_token", σημαίνει ότι ακολουθεί και επόμενη σελίδα απαντήσεων
        #(τα paths επισημαίνονται για debugging της διαδρομής)
        if 'next_token' in json_response['meta']:
            next_token = json_response['meta']['next_token']
            page_number = 1
            print("Περισσότερα tweets διαθέσιμα. Token επόμενης σελίδας: ", next_token)
            if result_count is not None and result_count > 0 and next_token is not None:
                print("Χρονικό διάστημα (path 1) από: ", start_time_list[i], " έως: ", end_time_list[i])
                print("Σελίδα:", page_number)
                append_to_csv(json_response, dataset)
                count += result_count
                total_tweets += result_count
                print("# από tweets που έχουν ληφθεί ως τώρα (path 1): ", total_tweets)
                print("\n")
                time.sleep(5) #μικρός χρόνος αναμονής για να μην λειτουργούμε καταχρηστικά

                #Ίδιος κώδικας όσο συνεχίζει να υπάρχει "next_tokens" στην απάντηση
                while next_token is not None:
                    json_response = connect_to_endpoint(url[0], headers, url[1], next_token)
                    result_count = json_response['meta']['result_count']
                    if 'next_token' in json_response['meta']:
                        next_token = json_response['meta']['next_token']
                        print("Περισσότερα tweets διαθέσιμα. Token επόμενης σελίδας: ", next_token)
                        if result_count is not None and result_count > 0 and next_token is not None:
                            page_number +=1
                            print("Χρονικό διάστημα (path 2) από: ", start_time_list[i], " έως: ", end_time_list[i])
                            print("Σελίδα:", page_number)
                            append_to_csv(json_response, dataset)
                            count += result_count
                            total_tweets += result_count
                            print("# από tweets που έχουν ληφθεί ως τώρα (path 2): ", total_tweets)
                            print("\n")
                            time.sleep(5)
```

```
            #Όταν δεν υπάρχει πια "new_tokens" έχουμε φτάσει στην τελευταία σελίδα
            else:
                print("Δεν υπάρχουν άλλα tweets. Τελευταία σελίδα για αυτό το χρονικό διάστημα.")
                if result_count is not None and result_count > 0:
                    page_number +=1
                    print("Χρονικό διάστημα (path 3) από: ", start_time_list[i], " έως: ", end_time_list[i])
                    print("Page:", page_number)
                    append_to_csv(json_response, dataset)
                    count += result_count
                    total_tweets += result_count
                    print("# από tweets που έχουν ληφθεί ως τώρα (path 3): ", total_tweets)
                    print("\n")
                    time.sleep(5)
                next_token = None
                flag = False #Τερματισμός του loop και μετακίνηση στο τυχόν επόμενο χρονικό διάστημα

        # Αν εξαρχής δεν υπάρχει "new_tokens", οι απαντήσεις χωράνε σε μια σελίδα
        else:
            print("Τα διαθέσιμα tweets για αυτό το χρονικό διάστημα χωράνε σε μία σελίδα.")
            if result_count is not None and result_count > 0:
                print("Χρονικό διάστημα (path 4) από: ", start_time_list[i], " έως: ", end_time_list[i])
                append_to_csv(json_response, dataset)
                count += result_count
                total_tweets += result_count
                print("# από tweets που έχουν ληφθεί ως τώρα (path 4): ", total_tweets)
                print("\n")
                time.sleep(5)
            flag = False
            next_token = None #Τερματισμός του loop και μετακίνηση στο τυχόν επόμενο χρονικό διάστημα

print("-------------------------------------")
print("-------------------------------------")
print("\n")
print("Τέλος. Συνολικός αριθμός tweets που αποθηκεύτηκαν στο .csv αρχείο: ", total_tweets)
```

A.2 Preparation

```
#Για εκτέλεση υπολογισμών σε πίνακες και λίστες
import pandas as pd
import numpy as np

#Για ανεύρευση προτύπων
import re

#Για διαχείριση .csv αρχείων
import csv

#Για διαχείριση html strings
import html

#Για χρήση εργαλείων επεξεργασίας φυσικής γλώσσας (Natural Language Toolkit)
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
```

```
from nltk.stem.wordnet import WordNetLemmatizer

#Για διόρθωση ορθογραφικών λαθών
from autocorrect import Speller

#Για εισαγωγή και διαχείριση έτοιμων datasets
from datasets import Dataset, load_dataset

#Για διαχείριση emoji και emoticons
import emoji
from emoji.unicode_codes import EMOJI_DATA

#nltk.download('all')
```

```
#Φόρτωμα μόνο του κειμένου των tweets σε ένα dataframe (ευέλικτο είδος πίνακα)

InputDataset='PoliticalTweets.csv'
OutputDataset='PoliticalTweets-Cleaned-All.csv'

#TweetData = pd.read_csv(InputDataset, delimiter='|', usecols=[3], on_bad_lines='skip')
TweetData = pd.read_csv(InputDataset, delimiter='|', usecols=[3], on_bad_lines='skip', nrows=20000)

print('Total number of tweets loaded in memory:', len(TweetData))

#Παροχή πληροφοριών για την μορφή του dataframe
#print(TweetData.shape)
#print(TweetData.info())
```

```
#Αφαίρεση links

#lambda: μικρή αδήλωτη συνάρτηση μίας εντολής, που εκτελείται επιτόπου
TweetData['Cleaned'] = TweetData['TweetBody'].apply(lambda x: re.sub(r"http\S+", "", x))

#Αφαίρεση του comment για προβολή των περιεχομένων του dataframe και επισκόπηση των αλλαγών
του βήματος.
#TweetData.head(50)
```

```
#Μετατροπή HTML χαρακτήρων διαφυγής

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: html.unescape(x))
```

```
# Αφαίρεση mentions

#Συνάρτηση εύρεσης κειμένου ορισμένου pattern και αφαίρεσής του
def FindRemovePattern(text, pattern):
 finds = re.findall(pattern, text)
 for i in finds:
    text = re.sub(i, '', text)
 return text

TweetData['Cleaned'] = np.vectorize(FindRemovePattern)(TweetData['Cleaned'], "@[\w]*")
```

```
#Σπασιμο φράσεων με underscores στις επιμέρους λέξεις

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: re.sub(r'[_]',' ',x))
```

```
#Συναρτήσεις αντικατάστασης emojis & emoticons με κείμενο
```

```
def convert_emojis(text):
    return(emoji.demojize(text).replace(':', ''))

def convert_emoticons(text):
    for emot in EMOTICONS:
        text = re.sub(u'('+emot+')', "_".join(EMOTICONS[emot].replace(",","").split()), text)
    return text

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: convert_emoticons(x))
```

```
#Αφαίρεση emoticons

def remove_emoticons(text):
    pattern = re.compile("["
    u"\U0001F600-\U0001F64F"  # Εικονίδια
    u"\U0001F300-\U0001F5FF"  # Σύμβολα
    u"\U0001F680-\U0001F6FF"  # Εικονίδια χαρτών/μετακινήσεων
    u"\U0001F1E0-\U0001F1FF"  # Σημαίες
    u"\U00002702-\U000027B0"
    u"\U000024C2-\U0001F251"
    u"\U0001f926-\U0001f937"
    u"\U00010000-\U0010ffff"
    "]+", flags=re.UNICODE)
    return pattern.sub(r'', text)

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: remove_emoticons(x))
```

```
# Αφαίρεση σημείων στίξης και συμβόλων (πχ hashtags)

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: re.sub(r'[^\w\s]',' ',x))
# r: εντός της αγκύλης υπάρχουν wildcards, όχι κείμενο
# ^:ο,τιδήποτε ΔΕΝ είναι | \w: λέξη | \s: κενό
```

```
# Αφαίρεση αριθμών και τυχόν υπόλοιπων μη αλφαβητικών χαρακτήρων

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: re.sub(r'[^a-zA-Zα-ωA-Ω
ΌΆΈΊΉΏάέίήόύώΐϊϋ]','',x))
```

```
#Αφαίρεση χαρακτήρων που ξέμειναν (και λέξεων ενός χαρακτήρα)

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: ' '.join([w for w in x.split() if len(w)>1]))
```

```
#Μετατροπή όλων των χαρακτήρων σε πεζούς

TweetData['Cleaned'] = TweetData['Cleaned'].apply(lambda x: x.lower())
```

```
#Σπάσιμο των φράσεων σε λίστες λέξεων, διευκολύνει τα επόμενα βήματα

TweetData['Tokenized'] = TweetData['Cleaned'].apply(lambda x: word_tokenize(x))
```

```
#Διόρθωση ορθογραφικών λαθών

#(Η ρύθμιση fast αναζητά μόνο ένα ορθογραφικό λάθος ανά λέξη (που είναι και το συνηθέστερο), αλλιώς
αργεί πάρα πολύ)

EnglishChecker = Speller(fast=True)
GreekChecker = Speller('el', fast=True)
```

```
def FixEnglish(words):
   return([EnglishChecker(word) for word in words])

def FixGreek(words):
   return([GreekChecker(word) for word in words])

TweetData['Tokenized'] = TweetData['Tokenized'].apply(lambda x: FixEnglish(x))
TweetData['Tokenized'] = TweetData['Tokenized'].apply(lambda x: FixGreek(x))
```

```
#Αφαίρεση stop-words (λέξεων που δεν περιέχουν συναισθηματική πληροφορία)

GreekSW = set(stopwords.words('greek'))
EnglishSW = set(stopwords.words('english'))

#print(GreekSW) #Προεπισκόπηση λεξικών
#print(EnglishSW)

#Εξαίρεση μεμονωμένων stop-words από την αφαίρεση
EnglishSW.remove('not')
GreekSW.remove('δεν')
GreekSW.remove('δε')
# Περισσότερες για εξαίρεση? =>'αλλα','μετά','παρά','τότε','ποτε','μετα','κατά','μη'

TweetData['Tokenized'] = TweetData['Tokenized'].apply(lambda x: [word for word in x if not word in
EnglishSW])
TweetData['Tokenized'] = TweetData['Tokenized'].apply(lambda x: [word for word in x if not word in
GreekSW])
```

```
#Λημματοποίηση (δλδ μετατροπή λέξεων στην βασική τους μορφή, στον βαθμό που υποστηρίζεται στην
γλώσσα μας) και επανένωση των λέξεων.

lemmatizing = WordNetLemmatizer()
TweetData['Final'] = TweetData['Tokenized'].apply(lambda x: ' '.join([lemmatizing.lemmatize(i) for i in x]))
```

```
#Αποθήκευση των επεξεργασμένων tweets από την μνήμη σε dataframe, και κατόπιν σε νέο .csv στον
δίσκο.

df = pd.DataFrame(TweetData['Final'])
#print(df)

df.to_csv(OutputDataset, index=False)
```

A.3 Tokenizer vocabulary

```
#Για αυτοματοποιημένη εισαγωγή παραμέτρων μοντέλων και tokenizers
from transformers import AutoModelForSequenceClassification, AutoTokenizer

#Για tokenization σε επίπεδο ολόκληρων λέξεων
import spacy

#Για εκτέλεση υπολογισμών σε πίνακες και λίστες
import numpy as np
import pandas as pd
```

```
#Για μετατροπή κειμένων σε πίνακες βαρύτητας βάσει συχνοτήτων εμφάνισης συγκεκριμένων strings
#(TF-IDF => term frequency-inverse document frequency)
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
#Download μοντέλου και tokenizer, και τοπική αποθήκευσή τους για γρηγορότερη πρόσβαση
#(μόνο για την πρώτη φορά)

model =
AutoModelForSequenceClassification.from_pretrained("MoritzLaurer/mDeBERTa-v3-base-mnli-xnli")
tokenizer = AutoTokenizer.from_pretrained("MoritzLaurer/mDeBERTa-v3-base-mnli-xnli", use_fast=False)

model.save_pretrained("C:/Users/Sifis/mymodels/mDeBERTa-v3-base-mnli-xnli")
tokenizer.save_pretrained("C:/Users/Sifis/mymodels/mDeBERTa-v3-base-mnli-xnli")
```

```
#Αρχικοποίηση του spaCY με χρήση του συνοδευτικού ελληνικού dataset

nlp = spacy.load("el_core_news_sm", exclude=['morphologizer', 'parser', 'ner', 'attribute_ruler', 'lemmatizer'])
```

```
#Συνάρτηση εκτέλεσης του tokenization

def spacy_tokenizer(document, nlp=nlp):
    doc = nlp(document)

    #Κάθε τι που περιλαμβάνεται μεταξύ δύο κενών και δεν είναι αλλαγή γραμμής, αποτελεί λέξη προς
επιστροφή
    tokens = [
        token.text for token in doc if (
        token.text.strip() != '' and \
        token.text.find("\n") == -1)]
    return tokens
```

```
%%time
#μέτρηση χρόνου εκτέλεσης του συγκεκριμένου κελιού

InputDataset='PoliticalTweets-Cleaned-Part.csv'

#Μετατρέπει το raw κείμενο σε πίνακα TF-IDF διανυσμάτων.
tfidf_vectorizer = TfidfVectorizer(lowercase=False, tokenizer=spacy_tokenizer,
                                   norm='l2', use_idf=True, smooth_idf=True, sublinear_tf=False)

# Μετατροπή του dataset σε λίστα συνεχόμενων προτάσεων, αφαιρώντας τις αλλαγές γραμμής
with open(InputDataset, 'r', encoding='utf-8') as file:
    docs = [line.rstrip('\n') for line in file]

length = len(docs)
result = tfidf_vectorizer.fit_transform(docs)
print(result.shape)

# Αποτέλεσμα:
# αριθμός προτάσεων-tweets / αριθμός συνολικών διαφορετικών λέξεων-tokens στο dataset
```

```
# Δημιουργία λιστών με τις συχνότητες εμφάνισης των λέξεων/tokens στο νέο dataset

def dfreq(idf, N):
    return (1+N) / np.exp(idf - 1) - 1

idf = tfidf_vectorizer.idf_  #Πίνακας αντίστροφων συχνοτήτων των λέξεων
```

```
#Αναγνωριστικά λέξεων sorted βάσει IDF
idf_sorted_indexes = sorted(range(len(idf)), key=lambda k: idf[k])

#Sorted πίνακας αντίστροφων συχνοτήτων
idf_sorted = idf[idf_sorted_indexes]

#Πίνακας tokens sorted βάσει συχνότητας
tokens_by_df = np.array(tfidf_vectorizer.get_feature_names_out())[idf_sorted_indexes]

#Sorted πίνακας αριθμού εμφανίσεων tokens
dfreqs_sorted = dfreq(idf_sorted, length).astype(np.int32)

#Sorted πίνακας tokens ΚΑΙ αριθμού εμφανίσεών τους
tokens_dfreqs = {tok:dfreq for tok, dfreq in zip(tokens_by_df,dfreqs_sorted)}

#Sorted πίνακας ποσοστιαίας εμφάνισης tokens
pct_list = [round(dfreq/length*100,9) for token,dfreq in tokens_dfreqs.items()]
```

```
# Φόρτωμα μοντέλου και tokenizer

model =
AutoModelForSequenceClassification.from_pretrained("C:/Users/Sifis/mymodels/mDeBERTa-v3-base-mnli-xnli")
tokenizer = AutoTokenizer.from_pretrained("C:/Users/Sifis/mymodels/mDeBERTa-v3-base-mnli-xnli/",
use_fast=False)
```

```
#Κώδικας για σύγκριση του λεξικού του αρχικού tokenizer με το νέο που δημιουργήσαμε από τα tweets.
#Κυρίως για εποπτικούς λόγους, δεδομένου ότι η addtokens() παραλείπει τις λέξεις που είναι ήδη γνωστές.

#0 για να περιλάβουμε κάθε διαθέσιμο token
pct = 0
index_max = len(np.array(tokens_pct_list)[np.array(pct_list)>=pct])
new_tokens = tokens_by_df[:index_max]

old_vocab = [k for k,v in tokenizer.get_vocab().items()]
new_vocab = [token for token in new_tokens]
idx_old_vocab_list = list()
same_tokens_list = list()
different_tokens_list = list()

for idx_new,w in enumerate(new_vocab):
 try:
   idx_old = old_vocab.index(w)
 except:
   idx_old = -1
 if idx_old>=0:
    idx_old_vocab_list.append(idx_old)
    same_tokens_list.append((w,idx_new))
 else:
    different_tokens_list.append((w,idx_new))

# Λέξεις του νέου μικρού λεξικού που:

print("Υπάρχουν ήδη στο αρχικό / καινούριες, δεν υπάρχουν στο αρχικό / συνολικές στο νέο λεξικό:")
len(same_tokens_list),len(different_tokens_list),len(same_tokens_list)+len(different_tokens_list)
```

```
#Εισαγωγή της λίστας νέων ολόκληρων λέξεων που πήραμε από τον spaCY, στον αρχικό tokenizer του μοντέλου

print("[ BEFORE ] tokenizer vocab size:", len(tokenizer))
added_tokens = tokenizer.add_tokens(new_tokens.tolist())

print("[ AFTER ] tokenizer vocab size:", len(tokenizer))
print()
print('added_tokens:',added_tokens)

# Αλλαγή του μεγέθους του πίνακα embeddings του μοντέλου
# για να ταιριάζει με το νέο αυξημένο μέγεθος του λεξικού του tokenizer
model.resize_token_embeddings(len(tokenizer))
```

```
#Τοπική αποθήκευση των εκτεταμένων μοντέλου-tokenizer

OutputModel="C:/Users/Sifis/mymodels/ExtendedAllDeBERTa"

model.save_pretrained(OutputModel)
tokenizer.save_pretrained(OutputModel)
```

Αξιολόγηση των εκτεταμένων μοντέλων (το προηγούμενο τμήμα του κώδικα δεν χρειάζεται)

```
from transformers import AutoModelForSequenceClassification, AutoTokenizer

TestModel="C:/Users/Sifis/mymodels/mDeBERTa-v3-base-mnli-xnli"
#TestModel="C:/Users/Sifis/mymodels/Extended1DeBERTa"
#TestModel="C:/Users/Sifis/mymodels/Extended001DeBERTa"
#TestModel="C:/Users/Sifis/mymodels/ExtendedAllDeBERTa"

model = AutoModelForSequenceClassification.from_pretrained(TestModel)
tokenizer = AutoTokenizer.from_pretrained(TestModel, use_fast=False)
```

```
TestText1="Το νομοσχέδιο για πρώτη φορά αναφέρεται συγκεκριμένα στους παρόχους υπηρεσιών ύδατος ως δημόσιους και δημοτικούς οργανισμούς, δεν ασχολείται σε κανένα άρθρο του με το ιδιοκτησιακό καθεστώς, ούτε με τη μετοχική σύνθεση των παρόχων υπηρεσιών ύδατος, η νέα Ρυθμιστική Αρχή έχει καθαρά εποπτικές και γνωμοδοτικές αρμοδιότητες, ενώ οι κανονιστικές αρμοδιότητες για την κοστολόγηση του νερού παραμένουν αρμοδιότητα των υπουργών και για τον σκοπό αυτόν εκδίδουν ΚΥΑ, είπε ο κ. Καππάτος."

#Δοκιμή του tokenization (encoding)

Tokens = tokenizer.tokenize(TestText1)
EncodedTokens = tokenizer(TestText1)
print("Αριθμός tokens =>", len(Tokens), "\n")
print(Tokens, "\n")
print(EncodedTokens)
```

```
#Δοκιμή της επανασύστασης του tokenized κειμένου (decoding)

tokenizer.decode(EncodedTokens['input_ids'], skip_special_tokens=True)
```

```
TestText2="Ο Τσίπρας είχε ψηφίσει νομοθεσία για υψηλή φορολογία και η ΝΔ την άλλαξε γιατί μπορούσε. Για τους πλειστηριασμούς επίσης ήθελε να τους αλλάξει αλλά δεν πρόλαβε.... Οπότε φταίει ο Τσίπρας. Δημοσκόπηση Marc: Οι επαναληπτικές εκλογές δείχνουν αυτοδυναμία της ΝΔ- Σημαντικό προβάδισμα 77 μονάδων με την απλή αναλογική. Η απόλυτα διάφανη στάση της κυβέρνησης και των ΜΜΕ στη συγκεκριμένη υπόθεση είναι χειροπιαστή απόδειξη ότι η ΝΔ εδώ και καιρό παίζει παιχνίδια με την ίδια τη δημοκρατία."
```

```
#Δοκιμή του tokenization (encoding)

Tokens = tokenizer.tokenize(TestText2)
EncodedTokens = tokenizer(TestText2)
print("Αριθμός tokens =>", len(Tokens), "\n")
print(Tokens, "\n")
print(EncodedTokens)
```

```
#Δοκιμή της επανασύστασης του tokenized κειμένου (decoding)

tokenizer.decode(EncodedTokens['input_ids'], skip_special_tokens=True)
```

A.4 Sentiment evaluation

```
#Για αυτοματοποιημένη εισαγωγή παραμέτρων μοντέλων και tokenizers
from transformers import AutoTokenizer, AutoModelForSequenceClassification

#Για εκτέλεση υπολογισμών σε πίνακες και λίστες
import pandas as pd

#Για διαχείριση .csv αρχείων
import csv

#Για επιτάχυνση των υπολογισμών με χρήση της GPU
import torch
```

```
#Αρχικοποίηση του μοντέλου και tokenizer

Model="C:/Users/Sifis/mymodels/mDeBERTa-v3-base-mnli-xnli/"

model = AutoModelForSequenceClassification.from_pretrained(Model)
tokenizer = AutoTokenizer.from_pretrained(Model, use_fast=False)

#Έλεγχος ύπαρξης GPU και προώθησης του μοντέλου σε αυτήν για επεξεργασία, αντί στην CPU
# (πολύ γρηγορότερο, ειδικά για μεγάλα datasets)
if torch.cuda.is_available():
    device = torch.device("cuda")
    model.to('cuda')
else:
    device = torch.device("cpu")

print("Device: ", device)
print(tokenizer.vocab_size)
```

```
#Ορισμός των συναισθημάτων που θέλουμε να διερευνήσουμε
#ως υποθέσεις (φράσεις νο2) που θα εξεταστούν διαδοχικά στα πλαίσια του NLI
#(γλώσσα και διατύπωση επηρεάζει τα αποτελέσματα)

hypothesis = ['χαρά', 'λύπη', 'θυμός', 'φόβος']

#Απόδοση των labels των αποτελεσμάτων στα ελληνικά
#(δεν επηρεάζει τα αποτελέσματα)
label_names = ["συμφωνία", "ουδετερότητα", "αντίθεση"]
```

```
# == Για αξιολόγηση ενός premise ==

premise = "Πάντα μέσα στην οικογένεια μου έβρισκα την αγάπη, την στήριξη, και την γαλήνη της ψυχής μου."
#premise = "Τίποτα πιο θλιβερό για τον κλασικό αθλητισμό, τον ΣΕΓΑΣ και την αθλητική ιστορία της προέδρου του."
#premise = "Αδιανόητα κακή η σημερινή εμφάνιση της ομάδας, ο προπονητής πρέπει να παραιτηθεί άμεσα!"
#premise = "Αγωνία για τον γνωστό ηθοποιό, σε κρίσιμη κατάσταση μετά από εγκεφαλικό."
#premise = "Τρόμος μέρα-μεσημέρι! Άνδρας με καλάσνικοφ άνοιξε πυρ στη μέση του δρόμου στην Αγία Βαρβάρα!"

#Δημιουργία κενών λιστών αποτελεσμάτων
lstent = []
lstcon = []

for i in hypothesis:    #Διαδοχική εκτέλεση για κάθε μία από τις υποθέσεις/συναισθήματα προς αξιολόγηση
    input = tokenizer(premise, i, truncation=True, return_tensors="pt")
    output = model(input["input_ids"].to(torch.device(device)))

    #Υπολογισμός συμφωνίας/ουδετερότητας/αντίθεσης
    prediction = torch.softmax(output["logits"][0], -1).tolist()

    #Αποθήκευση πιθανότητας συμφωνίας και αντίθεσης για την υπόθεση/συναίσθημα
    lstent.append(prediction[0])
    lstcon.append(prediction[2])

    #Εκτύπωση αποτελέσματος ως στρογγυλοποιημένο ποσοστό %
    results = {name: round(float(pred) * 100, 1) for pred, name in zip(prediction, label_names)}
    print(i, results)

#Index επικρατέστερου στην λίστα συναισθημάτων, ως index στις λίστες αποτελεσμάτων
index1 = lstent.index(max(lstent))
index2 = lstcon.index(min(lstcon))

print("\nΑποτέλεσμα βάσει μέγιστης συμφωνίας (",round(float(max(lstent)) * 100, 1),"% ) :", hypothesis[index1])
print("Αποτέλεσμα βάσει ελάχιστης αντίθεσης (",round(float(min(lstcon)) * 100, 1),"% ) :", hypothesis[index2])

#Υπολογισμός δείκτη αβεβαιότητας για κάθε λύση ως:
#(απόσταση συμφωνίας από το ιδανικό 100%) + (απόσταση αντίθεσης από το ιδανικό 0%)

vectent = round(((1-max(lstent)) + (lstcon[index1]))*100, 2)
vectcon = round(((1-lstent[index2]) + (min(lstcon)))*100, 2)

print("\n----------\n\nΓια αυτή την φράση =>",premise)
print("\nΔείκτης αβεβαιότητας λύσης maxent:", vectent)
print("Δείκτης αβεβαιότητας λύσης mincont:", vectcon)

if vectent<vectcon:
    print("\nΛιγότερο αμφίβολη πρόβλεψη =>", hypothesis[index1])
elif vectent>vectcon:
    print("\nΛιγότερο αμφίβολη πρόβλεψη =>", hypothesis[index2])
else:
    print("\nΟι δύο μέθοδοι συμφώνησαν στην πρόβλεψη =>", hypothesis[index1])
```

```
# == Για στατιστική αξιολόγηση ενός συνόλου από LABELED premises/tweets ==
# (με μορφή στηλών "Index | Text | Sentiment")

evalDF = pd.read_csv('LabeledTweets.csv', delimiter='|', usecols=[0,1,2], on_bad_lines='skip', encoding = 'ISO-8859-1')

#Δημιουργία του πίνακα αποθήκευσης των αποτελεσμάτων
```

```
#και αδειάσμά του σε περίπτωση που έχει δεδομένα από προηγούμενες εκτελέσεις του κώδικα

resultsDF = pd.DataFrame(columns=['MaxEnt', 'MinCont', 'BestGuess', 'LabeledSent', 'Success1',
'Success2', 'Success3'])
resultsDF = resultsDF[0:0]

for k in evalDF.index:    #Διαδοχική εκτέλεση για κάθε γραμμή του dataset (για κάθε tweet)

   premise = evalDF.at[k, 'TweetBody']

   #Δημιουργία κενών λιστών αποτελεσμάτων
   lstent = []
   lstcon = []

   for i in hypothesis:    #Διαδοχική εκτέλεση για κάθε μία από τις υποθέσεις/συναισθήματα προς
αξιολόγηση
      input = tokenizer(premise, i, truncation=True, return_tensors="pt")
      output = model(input["input_ids"].to(torch.device(device)))

      #Υπολογισμός συμφωνίας/ουδετερότητας/αντίθεσης
      prediction = torch.softmax(output["logits"][0], -1).tolist()

      #Αποθήκευση πιθανότητας συμφωνίας και αντίθεσης για την υπόθεση/συναίσθημα
      lstent.append(prediction[0])
      lstcon.append(prediction[2])

   #Index επικρατέστερου στην λίστα συναισθημάτων, ως index στις λίστες αποτελεσμάτων
   index1 = lstent.index(max(lstent))
   index2 = lstcon.index(min(lstcon))

   #Μετατροπή των INDEXES σε αριθμητικά συναισθήματα όπως στο .csv (+1 επειδή στα datasets
   #είναι συνήθως αριθμημένα ξεκινώντας από το 1, όχι από το 0 όπως τα indexes)
   #και αποθήκευση των συναισθημάτων, ώστε να είναι συγκρίσιμα με τα labeled.
   resultsDF.at[k, 'MaxEnt'] = index1+1
   resultsDF.at[k, 'MinCont'] = index2+1

   #Υπολογισμός δείκτη αβεβαιότητας για κάθε λύση ως:
   #(απόσταση συμφωνίας από το ιδανικό 100%) + (απόσταση αντίθεσης από το ιδανικό 0%)
   #και αποθήκευση της αντίστοιχης πρόβλεψης ως best guess.

   vectent = round(((1-max(lstent)) + (lstcon[index1]))*100, 2)
   vectcon = round(((1-lstent[index2]) + (min(lstcon)))*100, 2)

   if vectent<vectcon:
      resultsDF.at[k, 'BestGuess'] = resultsDF.at[k, 'MaxEnt']
   elif vectent>vectcon:
      resultsDF.at[k, 'BestGuess'] = resultsDF.at[k, 'MinCont']
   else:
      resultsDF.at[k, 'BestGuess'] = resultsDF.at[k, 'MaxEnt']

   resultsDF.at[k, 'LabeledSent'] = evalDF.at[k,'Sentiment']

   #Ελεγχος των αποτελεσμάτων σε σχέση με τα labeled sentiments
   #και αποθήκευση 1 ή 0 στις αντίστοιχες στήλες αποτελεσμάτων

   if resultsDF.at[k, 'MaxEnt'] == resultsDF.at[k, 'LabeledSent']:
      resultsDF.at[k, 'Success1']=1
```

```
    else:
        resultsDF.at[k, 'Success1']=0

    if resultsDF.at[k, 'MinCont'] == resultsDF.at[k, 'LabeledSent']:
        resultsDF.at[k, 'Success2']=1
    else:
        resultsDF.at[k, 'Success2']=0

    if resultsDF.at[k, 'BestGuess'] == resultsDF.at[k, 'LabeledSent']:
        resultsDF.at[k, 'Success3']=1
    else:
        resultsDF.at[k, 'Success3']=0

    print("\rΟλοκληρώθηκε η αξιολόγηση του tweet νο.:", k, end='', flush=True)

#resultsDF
print("\n----------\nAll done.\n\n")

#Εκτύπωση ποσοστών επιτυχίας προβλέψεων για κάθε τεχνική
print("Κάνοντας χρήση του μοντέλου =>", model.name_or_path,"\n")
print("- η τεχνική Maximum entailment είχε ποσοστό επιτυχίας:",
round(float(resultsDF['Success1'].sum()/resultsDF['Success1'].count())*100,2), "%")
print("- η τεχνική Minimum contradiction είχε ποσοστό επιτυχίας:",
round(float(resultsDF['Success2'].sum()/resultsDF['Success2'].count())*100,2), "%")
print("- η τεχνική ελάχιστης αβεβαιότητας είχε ποσοστό επιτυχίας:",
round(float(resultsDF['Success3'].sum()/resultsDF['Success3'].count())*100,2), "%")
```

```
# == Για στατιστική αξιολόγηση ενός συνόλου από UNLABELED premises/tweets ==
# (η στήλη των tweets πρέπει να έχει label "Final")

Dataset='4-Fear.csv'

#Encoding='utf-8'
Encoding='iso8859_7' #Αν δεν δουλέψει το UTF-8

evalDF = pd.read_csv(Dataset, delimiter='|', on_bad_lines='skip', encoding=Encoding, nrows=1000)

#Δημιουργία του πίνακα αποθήκευσης των αποτελεσμάτων
#και αδειάσμά του σε περίπτωση που έχει δεδομένα από προηγούμενες εκτελέσεις του κώδικα

resultsDF = pd.DataFrame(columns=['MaxEnt', 'MinCont', 'BestGuess'])
resultsDF = resultsDF[0:0]

for k in evalDF.index:    #Διαδοχική εκτέλεση για κάθε γραμμή του dataset (για κάθε tweet)

    premise = evalDF.at[k, 'Final']

    #Δημιουργία κενών λιστών αποτελεσμάτων
    lstent = []
    lstcon = []

    for i in hypothesis:    #Διαδοχική εκτέλεση για κάθε μία από τις υποθέσεις/συναισθήματα προς αξιολόγηση
        input = tokenizer(premise, i, truncation=True, return_tensors="pt") #PyTorch Tensors: πίνακες
        output = model(input["input_ids"].to(torch.device(device)))

        #Υπολογισμός συμφωνίας/ουδετερότητας/αντίθεσης
        prediction = torch.softmax(output["logits"][0], -1).tolist()

        #Αποθήκευση πιθανότητας συμφωνίας και αντίθεσης για την υπόθεση/συναίσθημα
```

```
        lstent.append(prediction[0]) #πχ συμφωνία: 0.9554
        lstcon.append(prediction[2]) #πχ αντίθεση: 0.0045

    #Index επικρατέστερου στην λίστα συναισθημάτων, ως index στις λίστες αποτελεσμάτων
    index1 = lstent.index(max(lstent)) #συναίσθημα συμφωνίας με την μέγιστη τιμή
    index2 = lstcon.index(min(lstcon)) #συναίσθημα διαφωνίας με την ελάχιστη τιμή

    #Αποθήκευση των επικρατέστερων συναισθημάτων για αυτό το tweet στα αποτελέσματα, ως κείμενο
    resultsDF.at[k, 'MaxEnt'] = hypothesis[index1]
    resultsDF.at[k, 'MinCont'] = hypothesis[index2]

    #Υπολογισμός δείκτη αβεβαιότητας για κάθε μία από τις δύο λύσεις ως:
    #(απόσταση συμφωνίας από το ιδανικό 100%) + (απόσταση αντίθεσης από το ιδανικό 0%)
    #και αποθήκευση της αντίστοιχης πρόβλεψης ως best guess.

    vectent = round(((1-max(lstent)) + (lstcon[index1]))*100, 2)
    vectcon = round(((1-lstent[index2]) + (min(lstcon)))*100, 2)

    if vectent<vectcon:
        resultsDF.at[k, 'BestGuess'] = resultsDF.at[k, 'MaxEnt']
    elif vectent>vectcon:
        resultsDF.at[k, 'BestGuess'] = resultsDF.at[k, 'MinCont']
    else:
        resultsDF.at[k, 'BestGuess'] = resultsDF.at[k, 'MaxEnt']

    print("\rΟλοκληρώθηκε η αξιολόγηση του tweet νο.:", k, end='', flush=True)

print("\n--------------\n\nAll done.\n\n--------------")
```

```
#Εκτύπωση preview του πίνακα αποτελεσμάτων
resultsDF.head(20)
```

```
#Δημιουργία και εκτύπωση πινάκων κατανομής υπολογισμένων συναισθημάτων στο dataset,
#για κάθε μια από τις χρησιμοποιούμενες τεχνικές.

results1 =
resultsDF.groupby("MaxEnt")["MaxEnt"].size().sort_values(ascending=False).reset_index(name='Counts')
results1['Counts'] = round(results1.Counts.div(results1.Counts.sum())*100,1)

results2 =
resultsDF.groupby("MinCont")["MinCont"].size().sort_values(ascending=False).reset_index(name='Counts')
results2['Counts'] = round(results2.Counts.div(results2.Counts.sum())*100,1)

results3 =
resultsDF.groupby("BestGuess")["BestGuess"].size().sort_values(ascending=False).reset_index(name='Counts')
results3['Counts'] = round(results3.Counts.div(results3.Counts.sum())*100,1)


print("\n== Αποτελέσματα ανάλυσης (",Dataset,") ==\n")

print("Ποσόστωση tweets ανά συναίσθημα\nβάσει max entailment (%) =>\n")
print(results1)

print("\nΠοσόστωση tweets ανά συναίσθημα\nβάσει min contradiction (%) =>\n")
print(results2)

print("\nΠοσόστωση tweets ανά συναίσθημα\nβάσει min δείκτη αβεβαιότητας (%) =>\n")
print(results3)
```