ΙΟΝΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ

Τμημα Πληροφορικης

IONIAN UNIVERSITY

Ανάλυση Συναισθημάτων comments του Stack Overflow χρησιμοποιώντας το μοντέλο SamLowe/roberta-base-go_emotions

Στέργιος Μουτζίκος
ΑΜ: inf2021149

Μάθημα: Σημασιολογικός και Κοινωνικός Ιστός
Διδάσκοντες: κ. Κορφιάτης, κα. Αίγλη

Ιούνιος 2025

Περιεχόμενα

.1 Περίληψη

Ο στόχος της παρούσας μελέτης είναι η ανάλυση των συναισθημάτων των χρηστών σε δεδομένα Κοινωνικού και Σημασιολογικού Ιστού με βάση το μοντέλο Transformer. Χρησιμοποιήθηκε το προ-εκπαιδευμένο μοντέλο **SamLowe/roberta-base-go_emotions** για να αναγνωριστούν 27 συναισθήματα σε σχόλια χρηστών. Στη μεθοδολογία περιλαμβάνεται μια αρχική φάση "εξερεύνησης", ακολουθούμενη από έναν εκτεταμένο προεπεξεργαστικό έλεγχο (αφαίρεση URLs/emojis, λημματοποίηση, ανάλυση συχνοτήτων λέξεων και bigrams) και την κατηγοριοποίηση metadata (φήμη χρήστη, αξιολογήσεις σχολίων κτλ).

Η ανάλυση έδειξε τη κατανομή των συναισθημάτων, συσχετίζοντάς τα με τη δημοτικότητα του περιεχομένου και με τα γεωγραφικά χαρακτηριστικά των χρηστών. Επιπρόσθετα, δίνεται έμφαση στα σχόλια με τη μεγαλύτερη συναισθηματική ακρίβεια και στη σχέση των συναισθημάτων με τα upvotes/downvotes. Τα ευρήματα αποτελούν πολύτιμη πηγή ενδείξεων για την πολυπλοκότητα των συναισθημάτων στο διαδίκτυο. Προτάσεις για μελλοντικές βελτιώσεις περιλαμβάνουν την προσθήκη δεδομένων από διαφορετικές χρονικές και περιβαλλοντικές διαστάσεις, καθώς και τη χρήση πιο εξειδικευμένων τεχνικών για την ανίχνευση περισσότερων συναισθημάτων.

.2 Εισαγωγή

Στη σύγχρονη ψηφιακή εποχή, όπου ο όγκος των παραγόμενων δεδομένων αυξάνεται εκθετικά, η αποκωδικοποίηση των ανθρώπινων συναισθημάτων και της ανθρώπινης έκφρασης έχει κεντρική σημασία. Ο Κοινωνικός και Σημασιολογικός Ιστός (Social and Semantic Web) διαθέτει τεράστιο όγκο πληροφοριών σχετικά με αυτά, με τα άρθρα στα social media, τις αναρτήσεις σε forum και blogs και τις κριτικές προϊόντων να αποτελούν απλώς ένα μέρος αυτών των πληροφοριών. Η Ανάλυση Συναισθημάτων (Sentiment Analysis), επίσης γνωστή ως Opinion Mining, αποτελεί έναν γρήγορα αναπτυσσόμενο κλάδο της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing - NLP) και της Τεχνητής Νοημοσύνης (Artificial Intelligence - AI) και έχει ως στόχο την αυτόματη ανίχνευση, ανάκτηση και κατηγοριοποίηση των συναισθημάτων που περιέχονται σε διάφορα κείμενα, όχι μόνο του Κοινωνικού Ιστού, αλλά και της Σημασιολογικής Ανάλυσης και γενικότερα του Διαδικτύου.

.2.1 Εισαγωγή στον Ερευνητικό Χώρο της Εργασίας

Η έρευνα που εξετάζει αυτή η εργασία ασχολείται με την ανάλυση πολύπλευρων συναισθημάτων στο Κοινωνικό Διαδίκτυο και τον Σημασιολογικό Ιστό. Συγκεκριμένα, η εργασία ασχολείται με τον τρόπο έκφρασης των συναισθημάτων σε σχόλια χρηστών και τον τρόπο με τον οποίο αυτά συνδέονται με πολλαπλά δεδομένα, όπως η δημοτικότητα μιας ανάρτησης, η φήμη του χρήστη που κάνει το σχόλιο, η γεωγραφική τοποθεσία και η απήχηση του σχολίου στην κοινότητα (π.χ. μέσω αναφορών, likes ή dislikes).

Η πολυπλοκότητα της ανθρώπινης γλώσσας, η ύπαρξη ειρωνείας, σαρκασμού, αλλά και η συνεχής εξέλιξη της ορολογίας και των εκφράσεων κάνουν την ανάλυση των συναισθημάτων δύσκολη. Η προσέγγιση της έρευνας στον Κοινωνικό Διαδίκτυο και τον Σημασιολογικό Ιστό είναι πιο λεπτομερής. Με τη χρήση ενός μοντέλου που μπορεί να αναγνωρίζει μια εκτεταμένη γκάμα συναισθημάτων (27 διαφορετικές κατηγορίες συναισθημάτων), η έρευνα καταφέρνει να κατανοήσει βαθύτερα τα συναισθήματα που βιώνουν οι χρήστες, να ξεπεράσει τους περιορισμούς των απλών προσεγγίσεων και να παρέχει πλουσιότερες πληροφορίες για την ανθρώπινη επικοινωνία στο διαδίκτυο. Η έρευνα αυτή δεν αφορά μόνο τα συναισθήματα, αλλά φθάνει μέχρι την ανάλυση των μοτίβων και των συνδέσεων που προκύπτουν ανάμεσα στους χρήστες και το περιεχόμενο.

.2.2 Βασικές Προσεγγίσεις Επίλυσής

Η ανάλυση συναισθημάτων έχει αναπτυχθεί σημαντικά μέσα στον χρόνο χρησιμοποιώντας διάφορες τεχνικές και μεθοδολογίες. Οι βασικές προσεγγίσεις διαχωρίζονται ως εξής:

1. **Προεπεξεργασία του Κειμένου (Text Preprocessing):** Πριν από την ανάλυση οποιουδήποτε είδους, τα un-edited κειμενικά δεδομένα απαιτούν κάποιες επεξεργασίες. Περιλαμβάνουν τα εξής:

    - **Μετατροπή σε πεζά γράμματα (Lowercasing:**Όλες οι λέξεις μετατρέπονται σε πεζά για να αντιμετωπιστούν ως ίδιες (π.χ., "Θυμός" και "θυμός").
    - **Αφαίρεση URLs και Emojis:**Οι ηλεκτρονικοί σύνδεσμοι και τα emojis, αν και μπορούν να μεταδώσουν συναισθήματα, συχνά προσθέτουν "θόρυβο" στην ανάλυση κειμένου ειδικά όταν χρησιμοποιούνται μοντέλα που βασίζονται στις λέξεις.Για την εστίαση στο λεκτικό περιεχόμενο, αυτά αφαιρούνται ή καθαρίζονται.

- **Αφαίρεση Stop Words:**Κοινές λέξεις που δεν προσθέτουν ουσιαστικό νόημα (π.χ., "ο", "η", "και", "είναι") αφαιρούνται για να μειωθεί ο όγκος των δεδομένων και να βελτιωθεί η εστίαση στις πιο σημαντικές λέξεις .

2. **Προσεγγίσεις Βασισμένες σε Μηχανική Μάθηση (Machine Learning-based Approaches):** Αυτές οι προσεγγίσεις χρησιμοποιούν μοντέλα μηχανικής μάθησης (π.χ. Support Vector Machines, Naive Bayes, Logistic Regression), τα οποία έχουν εκπαιδευτεί σε μεγάλα σύνολα δεδομένων με επισημασμένα τα συναισθήματα. Έτσι, το κείμενο μετατρέπεται σε αριθμητικά χαρακτηριστικά (π.χ. TF-IDF, Word Embeddings), τα οποία λαμβάνουν εκπαίδευση μέσω του μοντέλου. Η επίδοσή τους εξαρτάται κυρίως από τον τρόπο που διαμορφώνεται και το μέγεθος του όγκου της εκπαίδευσης. Η Η βελτίωση σε αυτόν τον τομέα οδήγησε στη δημιουργία βαθέων νευρωνικών δικτύων.

3. **Προσεγγίσεις Βασισμένες σε Βαθιά Μάθηση (Deep Learning-based Approaches):** Η εμφάνιση των σύγχρονων βαθιών νευρωνικών δικτύων, όπως τα Επαναλαμβανόμενα Νευρωνικά Δίκτυα (Recurrent Neural Networks - RNNs), τα Συνελικτικά Νευρωνικά Δίκτυα (Convolutional Neural Networks - CNNs) και κυρίως τα Transformers, ανοίγουν νέους ορίζοντες στην ανάλυση των συναισθημάτων. Αυτά τα μοντέλα μπορούν να δημιουργήσουν πολύπλοκες αναπαραστάσεις του κειμένου και να ανιχνεύουν τις μακροπρόθεσμες δημιουργούμενες, έτσι επιτυγχάνουν πολύ καλύτερες αποτελεσματικότητες. Τα μοντέλα Transformers, όπως το BERT και το RoBERTa, έχουν αποδειχθεί ιδιαίτερα αποτελεσματικά, διότι μπορούν να αντιληφθούν τον γενικό πλαίσιο των λέξεων σε μια πρόταση και να δημιουργήσουν υψηλής ποιότητας αναπαραστάσεις κειμένου (embeddings).

Η παρουσα εργασία εμπνέεται από μια μέθοδο βαθύς μάθησης, που πρόκειται για ένα μοντέλο Transformer (SamLowe/roberta-base-go emotions) που είναι προεκπαιδευμένο στην αναγνώριση πολλαπλών συναισθημάτων. Η συγκεκριμένη επιλογή δίνει τη δυνατότητα για την καλύτερη αντιμετώπιση των δυσκολιών που έχουν αναφερθεί, καθώς παρέχει μια λεπτομερή και ακριβή ανάλυση. Ο χρήστης μέσω ενός pipeline από τη βιβλιοθήκη Hugging Face καθιστά εύκολη την εκτέλεση αυτών των σύνθετων μοντέλων, δίνοντας την ευκαιρία να επικεντρωθεί στην ανάλυση των αποτελεσμάτων και όχι στην κατασκευή του μοντέλου.

.2.3 Συνεισφορά της Εργασίας

Η παρούσα εργασία συμβάλλει με λεπτομερειακό τρόπο στον τομέα της ανάλυσης συναισθημάτων, αναλυτικότερα:

1. **Πλήρης και Λεπτομερής Ανάλυση Συναισθημάτων:** Η εργασία δεν περιορίζε ται σε λίγες βασικές κατηγορίες, αλλά εφαρμόζει ένα προηγμένο μοντέλο για την αναγνώριση 27 διακριτών συναισθημάτων. Με αυτό το μοντέλο είναι εφικτό να γίνει πιο πλούσια και λεπτομερής κατανόηση των συναισθηματικών αποχρώσεων στα σχόλια των χρηστών που είναι το βασικό συστατικό για να βγουν ουσιαστικά συμπεράσματα.

2. **Συσχέτιση Συναισθημάτων με metadata:** Η εργασία δεν αναζητά μόνο τα συναισθήματα αλλά και συνδιάζει τα αποτελέσματα της ανάλυσης των συναισθημάτων με τα διάφορα μεταδεδομένα (όπως φήμη χρήστη, βαθμολογία ανάρτησης, βαθμολογία σχολίου, γεωγραφική τοποθεσία, upvotes/downvotes). Αυτή η η συγχώνευση των δεδομένων επιτρέπει τη διερεύνηση σύνθετων σχέσεων και μοτίβων, παρέχοντας απαντήσεις σε ερωτήματα όπως:

    - Πώς επηρεάζει η φήμη ενός χρήστη τα συναισθήματα που εκφράζει ή λαμβάνει;
    - Υπάρχουν συγκεκριμένα συναισθήματα που σχετίζονται με υψηλότερες ή χαμηλότερες βαθμολογίες αναρτήσεων/σχολίων;
    - Διαφέρουν τα εκφραζόμενα συναισθήσματα ανάλογα με την γεωγραφική τοποθεσία των χρηστών;
    - Πώς συσχετίζονται τα upvotes και downvotes με τα συναισθήματα που εκφράζονται στα σχόλια;
    -

3. **Οπτικοποίηση και Ερμηνεία των Αποτελεσμάτων:**Αναλυτικά γραφήματα (όπως ιστογράμματα, ραβδόγραμματα, heatmaps, box plots) αποτελούν σημαντικό τμήμα του έργου καθώς με τη βοήθειά τους γίνεται ευκολότερη η κατανόηση των πολύπλοκων δεδομένων και παρουσιάζονται έτσι τα ευρήματα με έναν πιο κατανοητό τρόπο τόσο στους ειδικούς όσο και στους μη ειδικούς.

4. **Χρήση Προηγμένων Εργαλείων NLP:** Η αξιοποίηση σύγχρονων εργαλείων NLP και βιβλιοθηκών (όπως οι transformers και το NLTK) και οι τεχνικές λημματοποίησης, εξαγωγής bigrams και το TF-IDF μοντέλο αποδεικνύουν ότι είμαστε σε θέση να εφαρμόσουμε προηγμένες μεθόδους στην επίλυση πραγματικών προβλημάτων.

5. **Εφαρμογές και Επιπτώσεις:** Τα ευρήματα αυτής της εργασίας μπορούν να έχουν σημαντικές πρακτικές εφαρμογές σε διάφορους τομείς, όπως:

    - Υπηρεσίες Πελατών: Κατανοώντας τα συναισθήματα των πελατών για τα προϊόντα ή τις υπηρεσίες που λαμβάνουν.
    - Μάρκετινγκ: Δίνοντας πληροφορίες σχετικά με την αντίληψη του κοινού για μια εταιρεία ή ένα προϊόν.
    - Κοινωνιολογία: Αναλύοντας κοινωνικές τάσεις και αντιδράσεις σε θέματα.

.3 Ερευνητικός χώρος (με βιβλιογραφική επισκόπηση)

Η έρευνα που παρουσιάζεται σε αυτή τη διατριβή εστιάζει στον τομέα όπου συναντιούνται η Ανάλυση Συναισθημάτων (Sentiment Analysis), η Επεξεργασία Φυσικής Γλώσσας (Natural Language Processing - NLP) και το Κοινωνικό και Σημασιολογικό Διαδίκτυο (Social and Semantic Web). Πρόκειται για έναν τομέα που έχει αποκτήσει τεράστια σημασία τα τελευταία χρόνια, καθώς ο όγκος του γραπτού υλικού που παράγεται από τους χρήστες στο διαδίκτυο έχει φτάσει σε εξωπραγματικά επίπεδα. Η δυνατότητα αυτόματης ανάλυσης και κατανόησης των συναισθημάτων που εκφράζονται σε αυτά τα δεδομένα είναι ζωτικής σημασίας για πολλούς τομείς, από την επιχειρηματική νοημοσύνη και το μάρκετινγκ έως την κοινωνιολογική έρευνα και την πρόβλεψη των τάσεων.

.3.1 Βιβλιογραφική Επισκόπηση

Η ανάλυση γνώμης, γνωστή και ως Opinion Mining, έχει βαθιές ρίζες στην επιστημονική κοινότητα. Αρχικά, η έρευνα επικεντρωνόταν στην κατηγοριοποίηση των κειμένων στις διακριτές κλιμακες της πολικότητας (θετικό, αρνητικό, ουδέτερο) [16] . Ωστόσο, η πολυπλοκότητα της ανθρώπινης γλώσσας και η ανάγκη για πιο λεπτομερή κατανόηση των συναισθημάτων οδήγησαν στην ανάπτυξη πιο προηγμένων τεχνικών, οι οποίες δίνουν τη δυνατότητα να αναγνωρίζουν μια ευρύτερη πληθώρα συναισθημάτων (όπως χαρά, λύπη, θυμός, έκπληξη, φόβος, εμπιστοσύνη) [2].

.3.1.1 Αρχικές Προσεγγίσεις

Η πρώτη φάση της ανάλυσης συναισθημάτων βασιζόταν κατά κύριο λόγο σε λεξικά συναισθημάτων, όπου οι λέξεις και οι φράσεις αντιστοιχίζονταν σε συγκεκριμένες συναισθηματικές

κλίμακες[11]. Παραδείγματα τέτοιων λεξικών περιλαμβάνουν το SentiWordNet [5], το οποίο αντιστοιχίζει σε σύνολα synsets από το WordNet συναισθηματικές τιμές και το AFINN [14], ένα λεξικό με βάση τις βαθμολογίες. Παρά το γεγονός ότι οι μέθοδοι αυτές είναι απλές και αποτελεσματικές για βασικού επιπέδου εφαρμογές, συχνά αποτυγχάνουν να συλλάβουν το πλήρες πλαίσιο της λέξης, τον σαρκασμό ή την ειρωνεία, καθώς και τις διαπολιτισμικές ιδιαιτερότητες της γλώσσας. [13].

.3.1.2 Προσεγγίσεις Μηχανικής Μάθησης

Με την αύξηση της ποσότητας των διαθέσιμων επισημασμένων δεδομένων, οι προσεγγίσεις βασισμένες σε μηχανική μάθηση (Machine Learning - ML) ξεπέρασαν τις υπόλοιπες. Παραδοσιακοί αλγόριθμοι όπως οι Support Vector Machines (SVMs)[9], Naive Bayes [15] και η Logistic Regression χρησιμοποιήθηκαν με επιτυχία για την κατηγοριοποίηση συναισθημάτων. Αυτές οι μέθοδοι απαιτούν τη μετατροπή του κειμένου σε αριθμητικές αναπαραστάσεις, όπως το Bag-of-Words ή το TF-IDF (Term Frequency-Inverse Document Frequency) [20]. Παρά τις προόδους, τα παραδοσιακά μοντέλα ML είχαν συχνά δυσκολίες να λύσουν το πρόβλημα της σημασιολογικής πολυπλοκότητας.

.3.1.3 Deep learning και Transformers

Η σημαντικότερη εξέλιξη που έχει σημειωθεί στον τομέα της επεξεργασίας φυσικής γλώσσας (NLP) και, κατ' επέκταση, στην αναγνώριση συναισθημάτων, είναι η είσοδος της Βαθιάς Μάθησης (Deep Learning - DL), και, συγκεκριμένα, των Transformer architectures [22]. Μοντέλα όπως το BERT (Bidirectional Encoder Representations from Transformers) [4], το RoBERTa (Robustly Opti- mized BERT Pretraining Approach)[12] και το GPT (Generative Pre-trained Transformer) [19] έχουν αλλάξει τον τρόπο με τον οποίο κατανοούμε το φυσικό λόγο. Αυτά τα μοντέλα, που έχουν υποστεί προεκπαίδευση σε τεράστια όγκου δεδομένων, μπορούν να κατανοήσουν τον συνολικό σημασιολογικό πυρήνα των λέξεων έτσι, ώστε να πετυχαίνουν πρωτοφανείς επιδόσεις σε πολλές εργασίες NLP, συμπεριλαμβανομένης της αναγνώρισης συναισθημάτων[23].

Το μοντέλο RoBERTa-base-go_emotions, το οποίο χρησιμοποιείται στην παρούσα έρευνα, ανήκει σε αυτή την κατηγορία. Πρόκειται για μια βελτιστοποιημένη έκδοση του BERT, η οποία έχει προσαρμοστεί για την ταξινόμηση συναισθημάτων σε 27 διακριτές κατηγορίες, όπως αυτές προκύπτουν από το σύνολο δεδομένων GoEmotions[3]. Αυτό το σύνολο δεδομένων περιλαμβάνει σχόλια από το Reddit και έχει σχεδιαστεί ειδικά για την ανίχνευση λεπτομερών

συναισθημάτων, προσφέροντας έτσι μια πιο λεπτομερή ανάλυση σε συναισθηματικό επίπεδο στα δεδομένα του[3].

.3.1.4 Ανάλυση Συναισθημάτων στον Κοινωνικό Ιστό

Τον Κοινωνικό Ιστό, δηλαδή τις πλατφόρμες όπως το Facebook, το X (πρώην Twitter), το Reddit και τα φόρα των συζητήσεων είναι ένας πλούσιος, αλλά και δύσκολος, χώρος για να αναλύσει κανείς τα συναισθήματα [1]. Τα δεδομένα που προέρχονται από αυτές τις πηγές έχουν τα εξής χαρακτηριστικά:

- **Ανεπίσημη γλώσσα:** Συχνή χρήση συντομογραφιών, αργκό, emojis και ανορθόγραφων λέξεων.
- **Πολυπλοκότητα κειμένου:**Το νόημα ενός σχολίου μπορεί να εξαρτάται από προηγούμενα σχόλια ή από τη γενική διάθεση της συζήτησης.
- **Θορυβώδη δεδομένα:**Παρουσία spam, διαφημίσεων ή άσχετου περιεχομένου.

Αρκετές έρευνες έχουν πραγματοποιηθεί για τη χρήση της ανάλυσης συναισθημάτων σε συγκεκριμένες πλατφόρμες κοινωνικών μέσων. Για παράδειγμα, η ανάλυση των tweets έχει εφαρμοστεί για τον προσδιορισμό των εκλογικών αποτελεσμάτων [21]και για τη μέτρηση της δημόσιας γνώμης για προϊόντα [7].Η ανάλυση των σχολίων σε φόρουμ ή πλατφόρμες όπως το Reddit παρέχει τη δυνατότητα να διερευνηθούν μακροσκελή άρθρα και πιο οργανωμένες συζητήσεις, αποκαλύπτοντας τις συναισθηματικές αντιδράσεις των χρηστών σε συγκεκριμένα θέματα ή αναρτήσεις [6].

.3.1.5 Συσχέτιση Συναισθημάτων με Χαρακτηριστικά Χρήστη

Εκτός από την απλή ανάλυση των συναισθημάτων, η σύγχρονη έρευνα επικεντρώνεται πλέον στην εξέταση του πώς τα συναισθήματα συνδέονται με άλλες συμπεριφορές και χαρακτηριστικά των χρηστών και του περιεχομένου. Έρευνες έχουν δείξει ότι η φήμη ενός χρήστη (π.χ., ο αριθμός των followers, το ιστορικό δημοσιεύσεών του) μπορεί να επηρεάσει τον τρόπο έκφρασης των συναισθημάτων του[8]. Αντίστοιχα, η δημοτικότητα μιας δημοσίευσης (όπως αυτή που μετρά likes, shares, upvotes) σχετίζεται συχνά με το συναισθηματικό πλαίσιο των σχολίων που λαμβάνει[10]. Η γεωγραφική θέση των χρηστών είναι άλλο ένα σημαντικό στοιχείο, καθώς οι διαφορετικές πολιτισμικές ομάδες μπορεί να έχουν διαφορετικούς τρόπους έκφρασης των συναισθημάτων τους [18]. Η ανάλυση των upvotes και downvotes στα σχόλια παρέχει

άμεσα στοιχεία για το αν το συναισθηματικό περιεχόμενο έχει γενική αποδοχή ή απορρίπτεται από τους χρήστες, και μας βοηθά να καταλάβουμε πώς κάποια συναισθήματα επηρεάζουν την κοινότητα [17].

.4 Το Μοντέλο SamLowe/roberta-base-go_emotions

Για την ανάλυση των συναισθημάτων, όπως εξηγήθηκε προηγουμένως, χρησιμοποιήθηκε το μοντέλο SamLowe/roberta- base-go_emotions. Πρόκειται για ένα προηγμένο εργαλείο βαθιάς μάθησης, βασισμένο στην αρχιτεκτονική Transformer, το οποίο αποτελεί έναν από τους ορόσημους στον τομέα της Επεξεργασίας Φυσικής Γλώσσας (NLP).

Το εν λόγω μοντέλο αποτελεί μια βελτιωμένη έκδοση του RoBERTa, το οποίο, από την άλλη, είναι μια περαιτέρω αναβάθμιση του BERT. Οι βελτιώσεις που έχει επιφέρει το RoBERTa περιλαμβάνουν το γεγονός ότι έχει εκπαιδευτεί σε πολύ πιο μεγάλα datasets και ότι χρησιμοποιεί μια πολύ πιο δυναμική στρατηγική "masking" λέξεων, γεγονός που το καθιστά πιο αποτελεσματικό και ισχυρό στην κατανόηση κάθε γλωσσικού πλαισίου. Το καθοριστικής σημασίας στοιχείο για την εργασία είναι ότι το εν λόγω μοντέλο έχει γίνει υπόλοιπο/διάπλαση (fine-tuned) με χρήση υπο-datasets που προέρχονται από το Stack Overflow Dataset. Το συγκεκριμένο σύνολο δεδομένων περιλαμβάνει σχόλια από το Stack Overflow τα οποία έχουν ταξινομηθεί σε 27 διαφορετικές κατηγορίες συναισθημάτων (π.χ., χαρά, θλίψη, θυμός, έκπληξη, ουδέτερο). Η διαδικασία αυτή επιτρέπει στο μοντέλο να αναγνωρίζει με πολύ μεγαλύτερη ακρίβεια μια ευρύτερη γκάμα συναισθημάτων, από το απλό θετικό/αρνητικό/ουδέτερο.

Η λειτουργία του:
Όταν εισάγεται ένα σχόλιο στο μοντέλο μέσω του pipeline της Hugging Face transformers:

- Το κείμενο διασπάται σε tokens.
- Αυτά τα tokens μετατρέπονται σε αριθμητικές αναπαραστάσεις (embeddings) που "συλλαμβάνουν" τη σημασία και το πλαίσιο.
- Οι Transformer στρώσεις επεξεργάζονται αυτές τις αναπαραστάσεις.

- Ένα τελικό επίπεδο ταξινόμησης παράγει πιθανότητες για κάθε ένα από τα 27 συναισθήματα.

Γιατί επιλέχθηκε:
Η επιλογή του SamLowe/roberta-base-go_emotions προσφέρει:

- Υψηλή ακρίβεια στην ανίχνευση συναισθημάτων. Λεπτομερής κατανόηση των συναισθηματικών αποχρώσεων, κάτι που είναι απαραίτητο για την πολύπλοκη ανάλυσή που πραγματοποιήται.
- Την ικανότητα να χειριστεί το πλαίσιο του κειμένου.
- Ευκολία χρήσης χάρη στην ενσωμάτωσή του στο Hugging Face App.
- Με αυτόν τον αλγόριθμο, διενεργήται μια βαθιά και αξιόπιστη ανάλυση των συναισθημάτων στα σχόλια του Κοινωνικού και Σημασιολογικού Ιστού.

.5 Μεθοδολογική Διαδικασία

Η παρούσα μελέτη ακολουθεί μια συστηματική μεθοδολογία για την ανάλυση συναισθημάτων σε δεδομένα Κοινωνικού και Σημασιολογικού Ιστού. Η διαδικασία περιλαμβάνει τρία βασικά στάδια: την περιγραφή των δεδομένων, την προεπεξεργασία αυτών και τη διεξαγωγή πειραμάτων μάθησης για την εξαγωγή συναισθημάτων και την περαιτέρω ανάλυσή τους.

.5.1 Περιγραφή Δεδομένων

Για τους σκοπούς αυτής της ανάλυσης, χρησιμοποιήθηκαν τρία υπό-datasets(comments.csv, users.csv, posts_answers.csv και το παραγόμενο comments_with_emotions.csv) του Stack Overflow Data Dataset, τα οποία φορτώθηκαν με τη χρήση της βιβλιοθήκης pandas.

- **comments.csv:** Αυτό το αρχείο περιέχει τα σχόλια των χρηστών. Αποτελεί το πρωτεύον σύνολο δεδομένων για την ανάλυση συναισθημάτων. Κατά την αρχική εξερεύνηση, ελέγχθηκε για missing-values, duplicates και η μοναδικότητα των user_id και post_id για να διασφαλιστεί η ακεραιότητα των δεδομένων. (Περιέχει στήλες όπως id, text, post_id, user_id και score).
- **posts_answers.csv:** Αυτό το αρχείο περιλαμβάνει δεδομένα σχετικά με τις αναρτήσεις στις οποίες ανήκουν τα σχόλια, καθώς και τις απαντήσεις τους. Ένα κρίσιμο

χαρακτηριστικό εδώ είναι η βαθμολογία (score) της ανάρτησης, η οποία υποδηλώνει τη δημοτικότητα ή τη χρησιμότητά της. (Περιέχει στήλες όπως id, comment_count, favorite_count και score).

- **users.csv:** Αυτό το αρχείο περιέχει πληροφορίες για τους χρήστες που αλληλεπιδρούν στην πλατφόρμα. Βασικά χαρακτηριστικά περιλαμβάνουν τη φήμη (reputation) του χρήστη, η οποία συχνά αντανακλά την επιρροή ή τη δραστηριότητά του στην κοινότητα, καθώς και την τοποθεσία (location) του. Επίσης, αναλύθηκε η κατανομή της φήμης και εντοπίστηκαν οι δέκα κορυφαίες τοποθεσίες χρηστών, παρέχοντας ένα γεωγραφικό πλαίσιο στην ανάλυση. (Περιέχει στήλες όπως id, display_name, location, reputation, up_votes και down_votes).

.5.2 Περιγραφή Προεπεξεργασίας

Η αποτελεσματικότητα οποιασδήποτε ανάλυσης κειμένου εξαρτάται σε μεγάλο βαθμό από την ποιότητα της προεπεξεργασίας των δεδομένων. Στην παρούσα εργασία, εφαρμόστηκε μια σειρά από βήματα 'καθαρισμού" και προετοιμασίας των κειμενικών δεδομένων, καθώς και μετασχηματισμών των αριθμητικών δεδομένων, ώστε να είναι κατάλληλα για την ανάλυση συναισθημάτων.

.5.2.1 Καθαρισμός Κειμένου (Text Cleaning)

Τα σχόλια των χρηστών, τα οποία συχνά περιέχουν "θόρυβο" από τον ανεπίσημο χαρακτήρα του διαδικτυακού λόγου, υποβλήθηκαν στους ακόλουθους μετασχηματισμούς:

- **Αφαίρεση URL και Emojis:** Εφαρμόστηκαν τακτικές (**URL_PATTERN, EMOJI_PATTERN**) για την αφαίρεση URLs και emojis, καθώς αυτά τα στοιχεία, αν και ενίοτε μεταφέρουν συναισθήματα, μπορούν να περιπλέξουν την επεξεργασία από τα μοντέλα NLP, ειδικά όταν η ανάλυση εστιάζει στο λεκτικό περιεχόμενο.
- **Μετατροπή σε Πεζά (Lowercasing):** Όλα τα γράμματα μετατράπηκαν σε πεζά για να διασφαλιστεί ότι οι ίδιες λέξεις αναγνωρίζονται ως τέτοιες ανεξαρτήτως κεφαλαιοποίησης (π.χ., "Excellent" και "excellent").
- **Φιλτράρισμα "meaningful" Κειμένου:** Χρησιμοποιήθηκε η συνάρτηση **has_meaningful_text** (η οποία βασίζεται στο **MEANINGFUL_TEXT_PATTERN**)

για να διασφαλιστεί ότι κάθε σχόλιο περιέχει τουλάχιστον μία λέξη με τρεις ή περισσότερους αλφαριθμητικούς χαρακτήρες, απορρίπτοντας σχόλια που αποτελούνται μόνο από σύμβολα ή πολύ σύντομες, ανούσιες εκφράσεις.

- **Αφαίρεση Διπλοτύπων και Μικρού Μήκους Σχολίων:** Τα διπλότυπα σχόλια αφαιρέθηκαν για να αποφευχθεί η υπερεκπροσώπηση συγκεκριμένων εκφράσεων. Επιπλέον, σχόλια με λιγότερες από τρεις λέξεις (**min_words=3**) απορρίφθηκαν, καθώς θεωρήθηκαν ότι παρέχουν ανεπαρκές περιεχόμενο για ουσιαστική συναισθηματική ανάλυση.

*Αυτά τα βήματα υλοποιήθηκαν μέσω των συναρτήσεων **clean_text** και **clean_comments**, οι οποίες εφαρμόστηκαν στο DataFrame των σχολίων.*

.5.2.2 Προετοιμασία Δεδομένων για Ανάλυση NLP

Για την περαιτέρω ανάλυση και εξαγωγή χαρακτηριστικών, εφαρμόστηκαν οι ακόλουθες τεχνικές NLP:

- **Tokenization:** Τα σχόλια διαχωρίστηκαν σε μεμονωμένες λέξεις (tokens) χρησιμοποιώντας τη συνάρτηση `word_tokenize` της βιβλιοθήκης `nltk`.
- **Αφαίρεση Stop Words:** Λέξεις που είναι κοινές στη γλώσσα και δεν προσθέτουν σημαντικό σημασιολογικό νόημα (π.χ., "the", "a", "is", "and") αφαιρέθηκαν χρησιμοποιώντας τη λίστα `stopwords` της `nltk`. Αυτό βοηθά στην εστίαση στις πιο ουσιαστικές λέξεις.
- **Λημματοποίηση (Lemmatization):** Κάθε λέξη αναχώρησε στην αρχική της λεξικογραφική μορφή (π.χ., "running", "ran" γίνονται "run") με τη χρήση του `WordNetLemmatizer`. Αυτό μειώνει την πολυπλοκότητα του λεξιλογίου και ομαδοποιεί τις σημασιολογικά παρόμοιες λέξεις. Τα λημματοποιημένα tokens αποθηκεύτηκαν στη στήλη `lemmatized_tokens`.
- **Εξαγωγή Bigrams:** Πέραν των μεμονωμένων λέξεων, αναλύθηκαν και τα *bigrams* (ζεύγη γειτονικών λέξεων), καθώς συχνά μεταφέρουν περισσότερο νόημα (π.χ., "not good"). Οι 20 πιο συχνές λέξεις και bigrams οπτικοποιήθηκαν.
- **TF-IDF Vectorization:** Για την ποσοτικοποίηση της σημασίας των λέξεων στα σχόλια, εφαρμόστηκε η διανυσματοποίηση `TF-IDF (Term Frequency-Inverse Document Frequency)` με χρήση του `TfidfVectorizer`. Ορίστηκαν `max_features=28000`,

`stop_words='english'` και `ngram_range=(1,1)` ώστε η ανάλυση να επικεντρωθεί σε unigrams.

.5.2.3 Κατηγοριοποίηση Αριθμητικών Δεδομένων (Binning)

Για να διευκολυνθεί η ανάλυση των συσχετίσεων μεταξύ των συναισθημάτων και των αριθμητικών χαρακτηριστικών, οι συνεχείς αριθμητικές στήλες κατηγοριοποιήθηκαν σε διακριτές ομάδες (*binning*) χρησιμοποιώντας τη συνάρτηση `pd.cut` μέσω της βοηθητικής συνάρτησης `bin_column`.

- **Ομαδοποίηση Φήμης Χρήστη (`rep_group`):** Η φήμη των χρηστών (`reputation`) χωρίστηκε σε πέντε κατηγορίες: "Very Low", "Low", "Medium", "High", "Very High".
- **Ομαδοποίηση Βαθμολογίας Ανάρτησης (`post_score_group`):** Η βαθμολογία της ανάρτησης (`score_post`) κατηγοριοποιήθηκε σε "Negative", "Low", "Moderate", "High", "Very High".
- **Ομαδοποίηση Βαθμολογίας Σχολίου (`comment_score_group`):** Η βαθμολογία του σχολίου (`score`) κατηγοριοποιήθηκε ομοίως σε "Negative", "Low", "Medium", "High", "Very High".

Αυτές οι κατηγοριοποιήσεις επέτρεψαν την οπτικοποίηση των κατανομών των συναισθημάτων σε σχέση με διαφορετικά επίπεδα φήμης και δημοτικότητας.

.5.3 Πειράματα Μάθησης

Τα πειράματα μάθησης επικεντρώθηκαν στην εφαρμογή του μοντέλου ανάλυσης συναισθημάτων και στην ενοποίηση των αποτελεσμάτων με τα metadata για την εξαγωγή ουσιαστικών συμπερασμάτων.

.5.3.1 Ανάλυση Συναισθημάτων με GoEmotions Model

Η βασική ανάλυση συναισθημάτων, όπως αναφερθηκε και προηγουμένος, πραγματοποιήθηκε χρησιμοποιώντας το προεκπαιδευμένο μοντέλο `SamLowe/roberta-base-go_emotions` από τη βιβλιοθήκη `transformers` της Hugging Face.

- **Εφαρμογή Μοντέλου:** Το μοντέλο φορτώθηκε ως `text-classification pipeline` με `top_k=None` για την εξαγωγή πιθανοτήτων για όλες τις 27 κατηγορίες συναισθημάτων.
- **Batch Inference:** Για την επιτάχυνση της πρόβλεψης σε μεγάλο όγκο σχολίων, χρησιμοποιήθηκε η συνάρτηση `batch_infer` με `batch_size=16`. Κάθε σχόλιο περικόπηκε στα 512 tokens, το μέγιστο επιτρεπτό από το μοντέλο.
- **Αποθήκευση Αποτελεσμάτων:** Τα προβλεπόμενα συναισθήματα αποθηκεύτηκαν στη στήλη `emotions` του `DataFrame comments` και εξήχθησαν σε αρχείο `comments_with_emotions.csv` για μελλοντική χρήση.

.5.3.2 Εξαγωγή και Δομή Δεδομένων Συναισθημάτων

Αφού ανιχνεύθηκαν τα συναισθήματα, ακολούθησαν τα εξής βήματα:

- **Φιλτράρισμα με Thresholding:** Η συνάρτηση `extract_labels` εφάρμοσε κατώφλι `threshold=0.3` για να διατηρηθούν μόνο οι ετικέτες με υψηλή βεβαιότητα στη στήλη `emotion_labels`.
- **"Explode" Πολλαπλών Συναισθημάτων:** Χρησιμοποιήθηκε η μέθοδος `explode("emotion_labels")` ώστε κάθε συναίσθημα να αποτελεί ξεχωριστή γραμμή, διευκολύνοντας την ανάλυση συχνοτήτων και συσχετίσεων.

.5.3.3 Συγχώνευση Δεδομένων για Ανάλυση

Για τη μελέτη των σχέσεων μεταξύ συναισθημάτων και μεταδεδομένων, τα επεξεργασμένα σχόλια συγχωνεύτηκαν με τα δεδομένα χρηστών και αναρτήσεων.

- **Ενοποίηση DataFrames:** Η συνάρτηση `merge_all` εφάρμοσε *left merge* του `comments_exp` με τα `users` και `posts`, με βάση τα `user_id` και `post_id`. Πριν τη συγχώνευση, τα `user_id` και `post_id` μετατράπηκαν σε αριθμητικό τύπο για αποφυγή σφαλμάτων.

Αυτή η συγχωνευμένη δομή αποτέλεσε τη βάση για όλες τις περαιτέρω αναλύσεις και οπτικοποιήσεις, επιτρέποντας τη διερεύνηση σύνθετων σχέσεων μεταξύ εκφρασμένων συναισθημάτων και χαρακτηριστικών χρηστών και περιεχομένου.

.6 Ανάλυση και Αξιολόγηση

Η φάση της ανάλυσης και αξιολόγησης επικεντρώνεται στην εξαγωγή και ερμηνεία των συναισθημάτων που προκύπτουν από την ανάλυση κειμένων, καθώς και στη διερεύνηση των συνδέσεων μεταξύ αυτών των συναισθημάτων και των χαρακτηριστικών των χρηστών και των δημοσιεύσεών τους. Μιας και χρησιμοποιήθηκε ένα προ-εκπαιδευμένο μοντέλο για την ανίχνευση των συναισθημάτων, η αξιολόγηση επικεντρώνεται στην ποιότητα των πληροφοριών που ανακτήθηκαν από τα δεδομένα και όχι στην απόδοση του μοντέλου. Η ανάλυση γίνεται πιο κατανοητή μέσω διαφόρων γραφημάτων και διαγραμμάτων.

.6.1 Εξερεύνηση και Χαρακτηριστικά Δεδομένων

.6.1.1 Επισκόπηση Συνόλων Δεδομένων

- **Σχόλια (comments)**: Έγινε έλεγχος των τίτλων, των κενών τιμών (missing values) και των διπλο-αναρτήσεων. Έτσι, προέκυψε ο συνολικός αριθμός των σχολίων αλλά και ο αριθμός των μοναδικών χρηστών και των αναρτήσεων.

- **Αναρτήσεις (posts)**: Σε αντίστοιχη διαδικασία, εξετάστηκαν οι τίτλοι, τα missing values και τα διπλο-αναρτήσεις. Επιπλέον, δόθηκε έμφαση στη διανομή των βαθμολογιών.

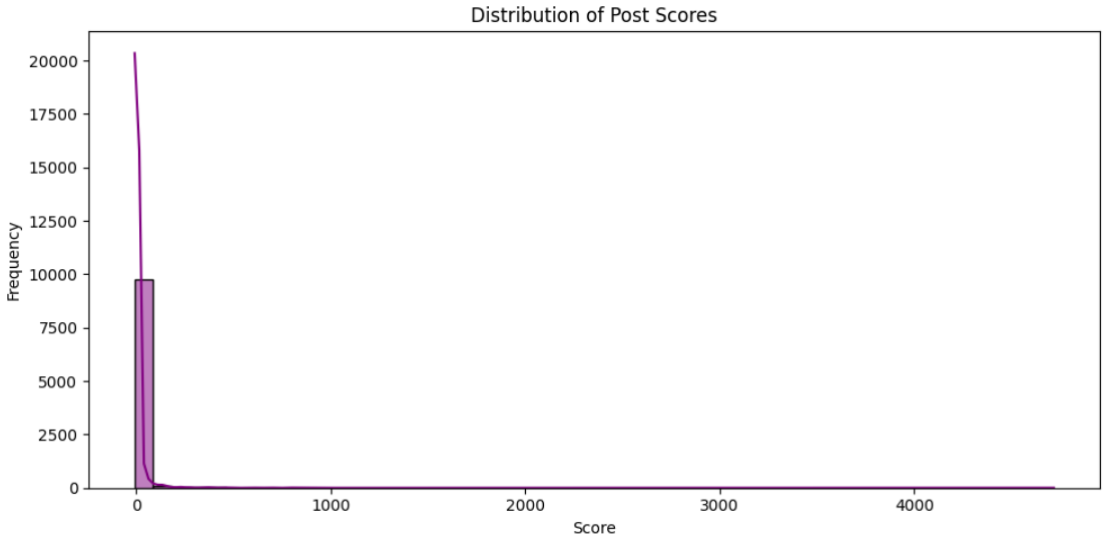

Figure 1: *Distribution of post scores*

- **Χρήστες (users)**: Ελέγχθηκαν οι επικεφαλίδες των στηλών, τα κενά και τα διπλότυπα των δεδομένων. Αναλύθηκε η κατανομή της φήμης των χρηστών και ορίστηκαν οι δέκα κορυφαίες τοποθεσίες.

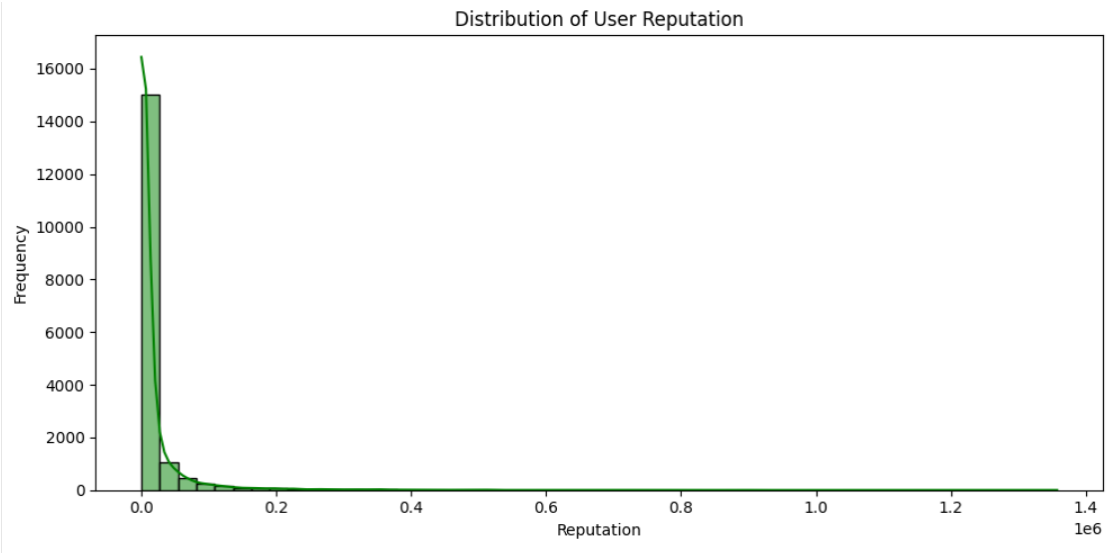

Figure 2: *Distribution of user reputation*

```
Top 10 User Locations:
location
India                          248
Germany                        230
United States                  174
United Kingdom                 172
London, United Kingdom         159
Berlin, Germany                106
Netherlands                    106
France                          99
Bangalore, Karnataka, India     89
Paris, France                   77
Name: count, dtype: int64
```

Figure 3: *Top 10 user locations*

.6.1.2 Χαρακτηριστικά Κειμένου των Σχολίων

- **Κατανομή Μήκους Σχολίων**: Ο υπολογισμός του αριθμού των λέξεων σε κάθε σχόλιο και η οπτικοποίησή του.

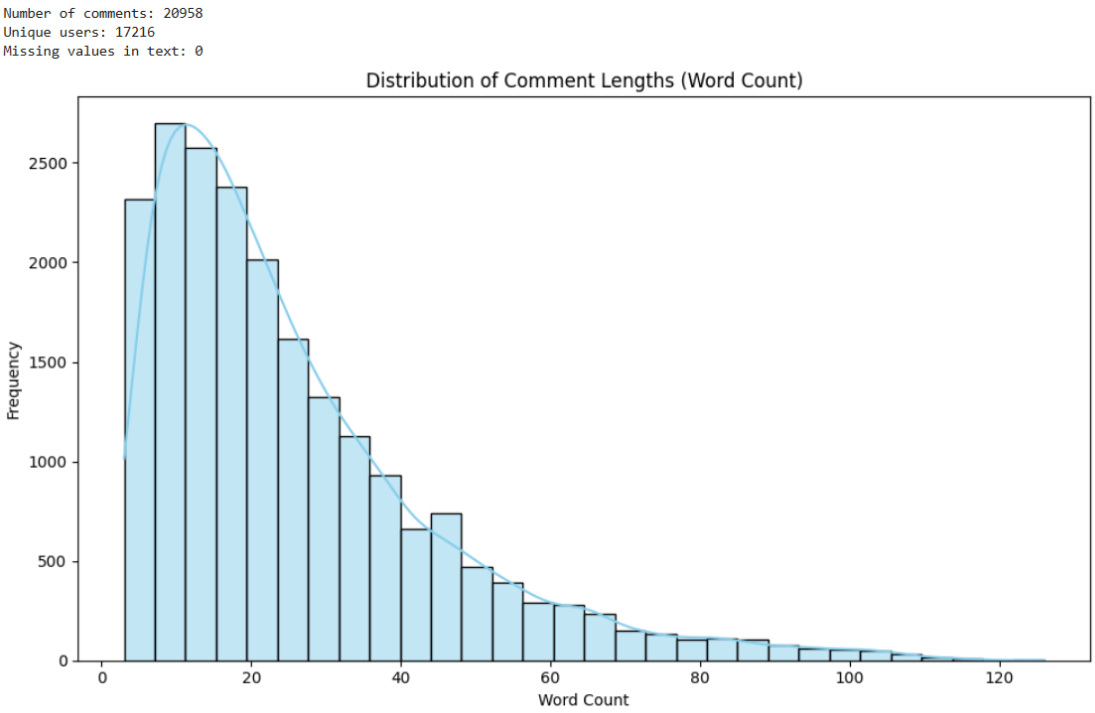

Figure 4: *Word count*

- **Ανάλυση Συχνότητας Λέξεων και Bigrams**: Προσδιορίστηκαν οι πιο συχνές λέξεις και φράσεις.

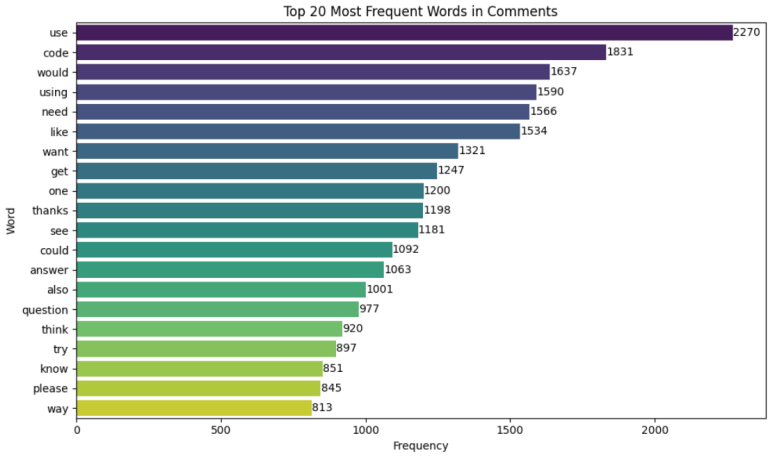

Figure 5: *Συχνότητα λέξεων*

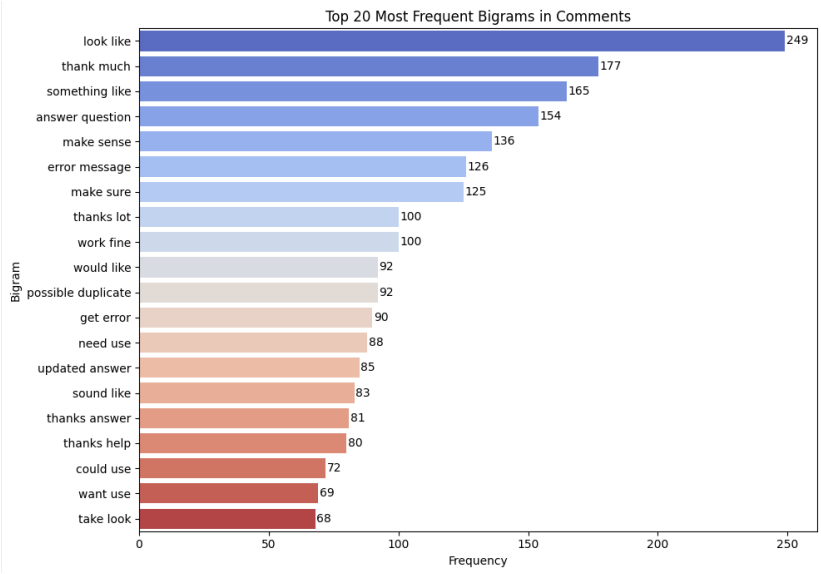

Figure 6: *Most frequent bigrams*

.6.1.3 Γενική Κατανομή Συναισθημάτων

Αρχικά, εξετάστηκε η συνολική κατανομή των 27 διακριτών συναισθημάτων που ανιχνεύθηκαν στα σχόλια.

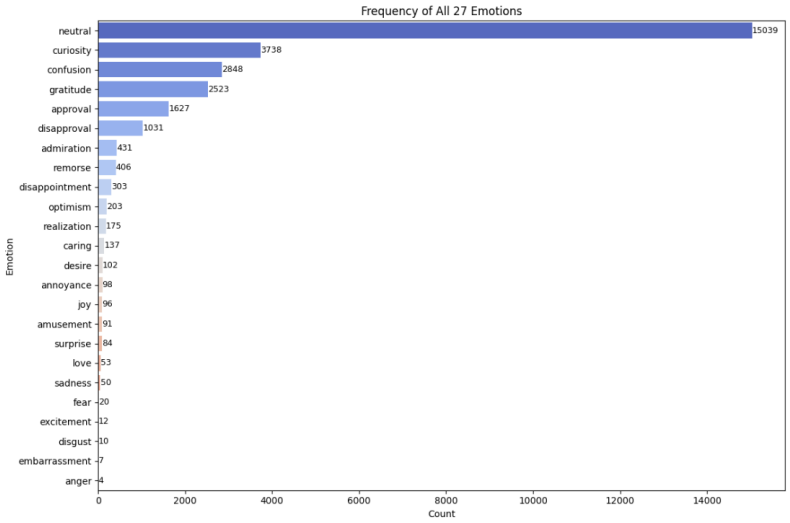

Figure 7: *Frequency of All 27 Emotions (Bar Plot)*

Το διάγραμμα παρουσιάζει ποια συναισθήματα εκφράζονται περισσότερο σε όλα τα σχόλια.

Δίνει μια ιδέα για το ποια συναισθήματα είναι πιο δημοφιλή στις Stack Overflow συζητήσεις. Υπολογίζεται ότι το άκρως αντικειμενικό συναίσθημα θα είναι ανάμεσα στα πιο συχνά, καθώς πολλά σχόλια μπορεί να μην έχουν έντονο συναισθηματικό φορτίο ή να είναι απλά πληροφοριακά.

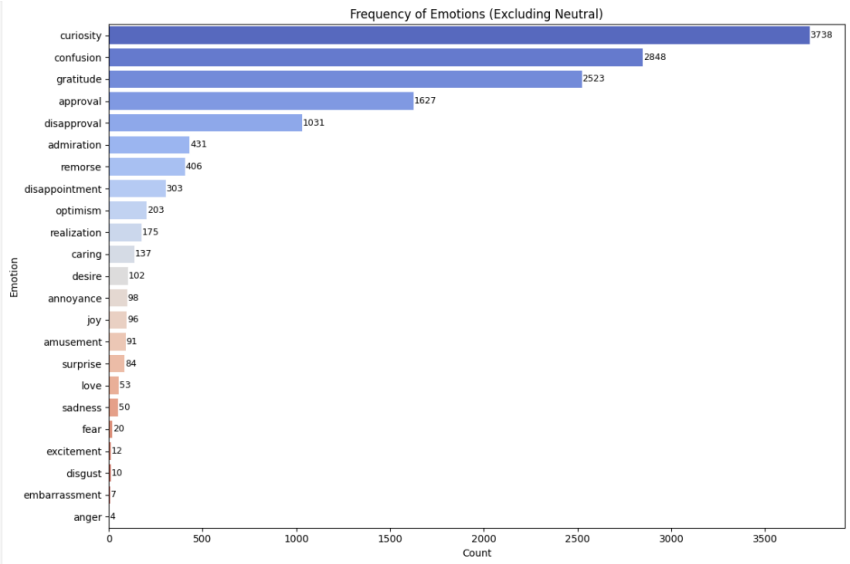

Figure 8: *Frequency of Emotions (Excluding Neutral)*

Αυτό το κείμενο αποκλείει την κατηγορία ουδέτερο, καθώς φανερώνει με κυρίαρχο τρόπο θετικά, αρνητικά ή ακόμα και ανάμεικτα συναισθήματα. Για παράδειγμα, η συχνή αναφορά στην ευγνωμοσύνη ή την έγκριση μπορεί να υποδηλώνει μια γενικά θετική ατμόσφαιρα, ενώ η αναφορά στη δυσαρέσκεια μπορεί να αποτελεί ένδειξη έντασης.

.6.1.4 Σχέση Συναισθημάτων με τη Δημοτικότητα του Περιεχομένου

Μελετήθηκε η σχέση των συναισθημάτων που εκφράζονται στα σχόλια με τη δημοτικότητά τους και τη δημοτικότητα των δημοσιεύσεων, στις οποίες ανήκουν. Τα σκορ (scores) ταξινομήθηκαν σε κατηγορίες (Negative, Low, Medium, High, Very High).

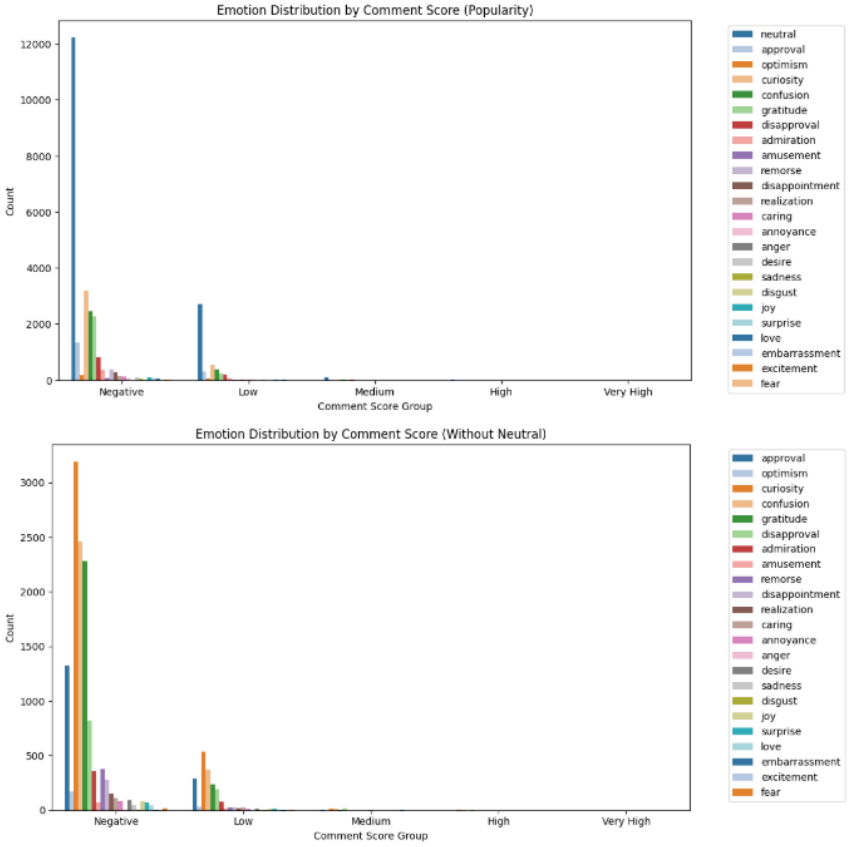

Figure 9: *Emotion Distribution by Comment Score*

Το παραπάνω γράφημα δείχνει την κατανομή συναισθημάτων σε σχέση με τη βαθμολογία των σχολίων.

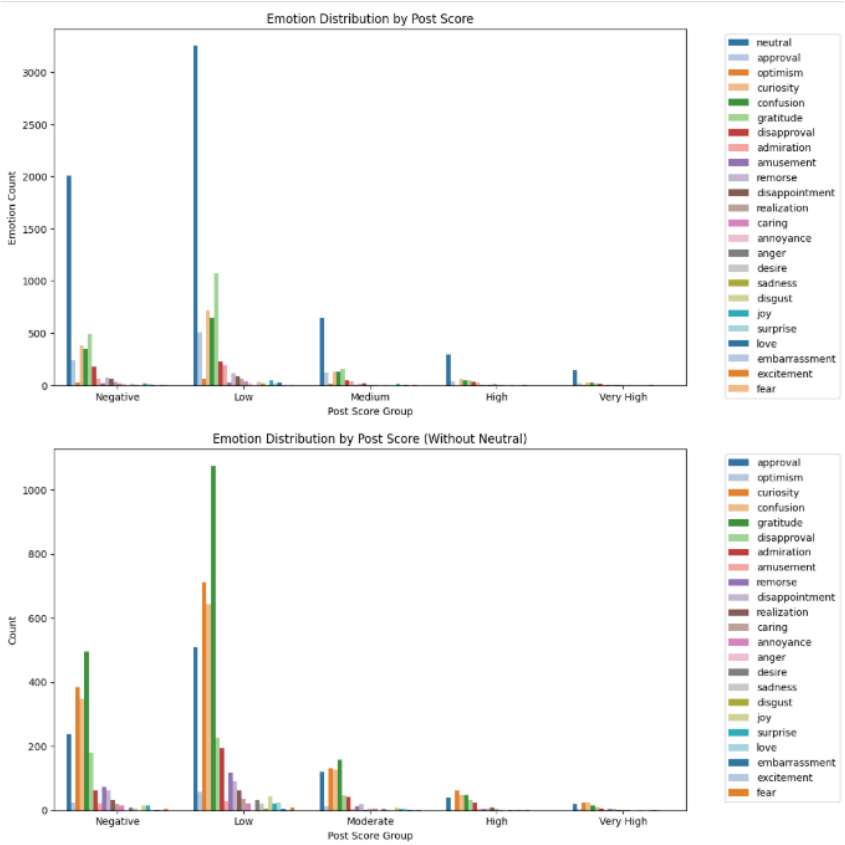

Figure 10: *Emotion Distribution by Post Score*

Αντίστοιχα, αυτό το γράφημα δείχνει την κατανομή συναισθημάτων σε σχέση με τη βαθμολογία των αναρτήσεων στις οποίες ανήκουν τα σχόλια.

.6.1.5 Γεωγραφικές Διαφοροποιήσεις ανά συναίσθημα

Αναζητήθηκαν διαφορές στην έκφραση συναισθημάτων ανάλογα με τη γεωγραφική τοποθεσία των χρηστών, με εστίαση στις 10 κορυφαίες περιοχές.

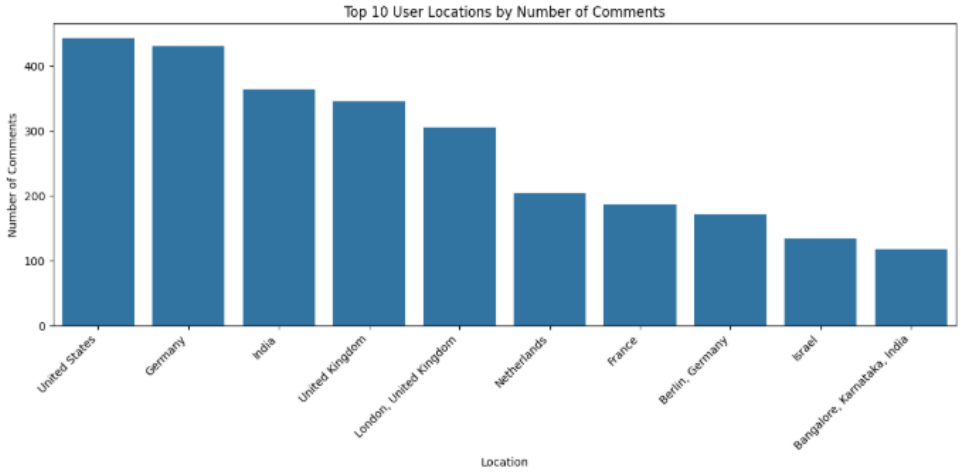

Figure 11: *Emotions by User Location (Top 10 Locations)*

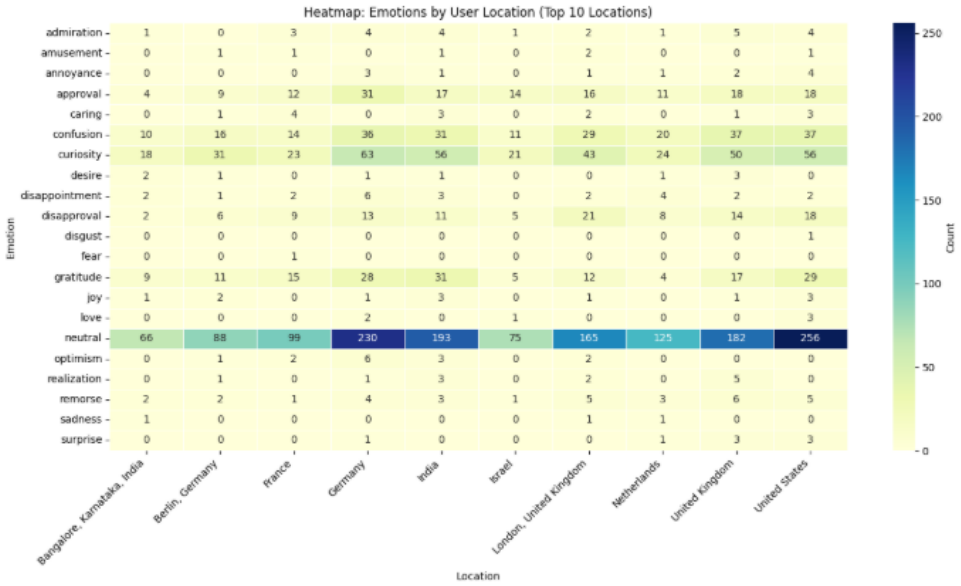

Figure 12: *Emotions Heatmap by User Location*

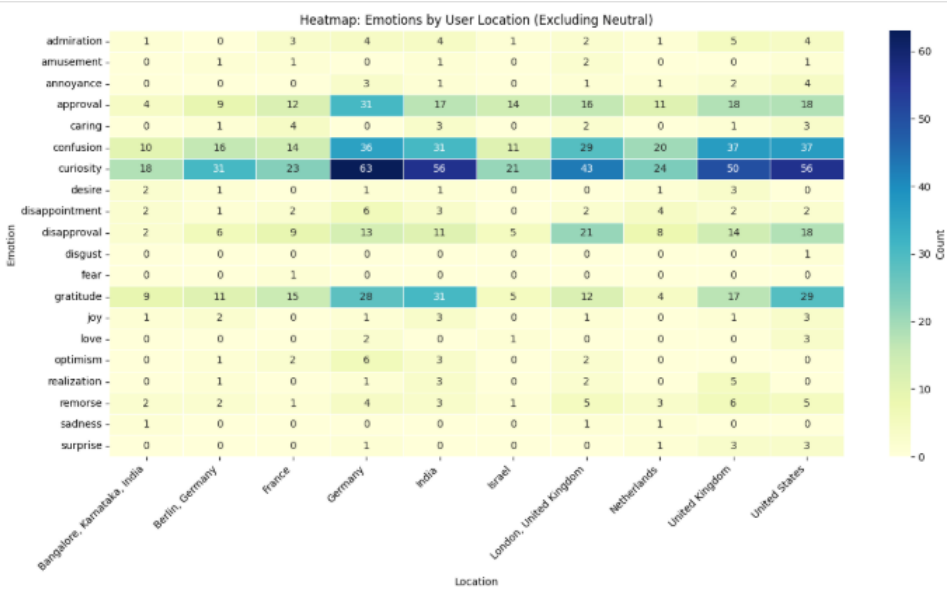

Figure 13: *Emotions Heatmap by User Location (Excluding Neutral)*

.6.1.6 Συναισθήματα και Αντίδραση της Κοινότητας

Η ανάλυση εξετάζει τη σχέση μεταξύ συναισθημάτων και της αντίδρασης της κοινότητας (upvotes/downvotes):

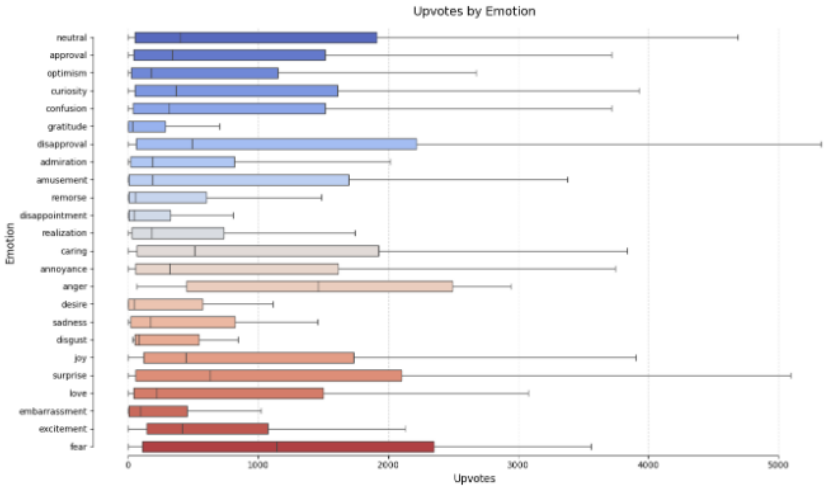

Figure 14: *Upvotes by Emotion (Box Plot)*

Ποιες συναισθηματικές εκφράσεις τείνουν να επιβραβεύονται με περισσότερα upvotes;

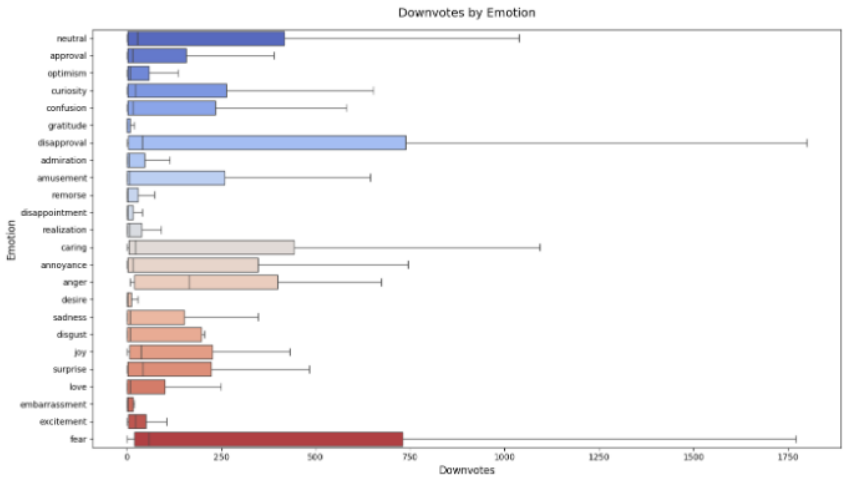

Figure 15: *Downvotes by Emotion (Box Plot)*

Αντίστοιχα, ποιες συναισθηματικές εκφράσεις τείνουν να προκαλούν αρνητική ανατροφοδότηση;

Διαβάζοντας προσεκτικά αυτήν την ανάλυση δεδομένων και με βάση τις οπτικοποιήσεις, μπορούμε να καταλήξουμε σε σημαντικά συμπεράσματα σχετικά με τον τρόπο έκφρασης των συναισθημάτων στον διαδίκτυο (Κοινωνικός και Σημασιολογικός Ιστός) και τον τρόπο που σχετίζεται με τα χαρακτηριστικά των χρηστών και το περιεχόμενο.

.7 Συμπεράσματα

Η έρευνα αυτή επέδειξε την ισχύ των μοντέλων Transformer, όπως το *SamLowe/roberta-base-go_emotions*, σε λεπτομερή αναλύσεις συναισθημάτων που βασίζονται σε δεδομένα κοινωνικού δικτύου. Χάρη στην ανάλυση, καταφέρθηκε να απεικονιστεί ο τρόπος διάχυσης των 27 διαφορετικών συναισθημάτων στα σχόλια των χρηστών, αποκαλύπτοντας έτσι τον πλούτο των συναισθημάτων. Διαπιστώθηκε η σχέση των συναισθημάτων με τη δημοτικότητα των σχολίων και των δημοσιεύσεων, καθώς και τα γεωγραφικά χαρακτηριστικά των χρηστών. Η έρευνα σχετικά με τα κορυφαία σχόλια και τις προβλέψεις που πραγματοποιήθηκαν με υψηλή επιτυχία πρόσφερε μια ολοκληρωμένη εικόνα της απόδοσης του μοντέλου. Επίσης, η ανάλυση των upvotes και downvotes ανά συναίσθημα ήταν πολύτιμη για την κατανόηση της κοινοτικής αντίδρασης σε συγκεκριμένες συναισθηματικές εκφράσεις. Όλα αυτά τα ευρήματα έχουν μεγάλη σημασία για την κατανόηση της δυναμικής της επικοινωνίας στις διαδικτυακές πλατφόρμες.

.8 Προτάσεις για Μελλοντικές Βελτιώσεις

Βασιζόμενοι στα ευρήματα και την εμπειρία από την παρούσα μελέτη, προτείνονται οι ακόλουθες βελτιώσεις και επεκτάσεις για μελλοντική έρευνα:

.8.1 Εμπλουτισμός Δεδομένων

- **Ενσωμάτωση Μεταδεδομένων Χρόνου**: Η ανάλυση της χρονικής διάστασης (π.χ. η ώρα της ημέρας, η ημέρα της εβδομάδας, εποχικές τάσεις) θα μπορούσε να αποκαλύψει ενδιαφέροντα μοτίβα στην έκφραση συναισθημάτων.
- **Πληροφορίες Θέματος/Κατηγορίας Ανάρτησης**: Η κατηγοριοποίηση των αναρτήσεων ανά θέμα θα επέτρεπε τη διερεύνηση συναισθηματικών διαφορών μεταξύ διαφορετικών γνωστικών πεδίων ή ενδιαφερόντων.
- **Ανάλυση Συνομιλίας (Conversational Context)**: Αντί της ανάλυσης μεμονωμένων σχολίων, η εξέταση ολόκληρων νημάτων συζητήσεων θα μπορούσε να αποκαλύψει τη συναισθηματική εξέλιξη των αλληλεπιδράσεων.

.8.2 Ποιοτική Ανάλυση & Επαλήθευση

- **Συνέντευξη Χρηστών**: Ποιοτική έρευνα μέσω συνεντεύξεων με χρήστες θα μπορούσε να παρέχει βαθύτερες γνώσεις για το γιατί εκφράζουν συγκεκριμένα συναισθήματα και πώς αντιλαμβάνονται τις αντιδράσεις της κοινότητας.

.8.3 Προηγμένες Τεχνικές Ανάλυσης

- **Μοντέλα Πολλαπλών Γλωσσών**: Για πλατφόρμες με πολλαπλές γλώσσες, η χρήση πολυγλωσσικών μοντέλων Transformer θα επέτρεπε την ανάλυση συναισθημάτων χωρίς περιορισμό στη γλώσσα.
- **Ανίχνευση Σαρκασμού/Ειρωνείας**: Η ενσωμάτωση ειδικών μοντέλων ή τεχνικών για την ανίχνευση σαρκασμού και ειρωνείας θα βελτίωνε σημαντικά την ακρίβεια της συναισθηματικής ανάλυσης.

.9 Βιβλιογραφία

[1] Charu C. Aggarwal. *Social Network Data Analytics*. Springer, 2011.

[2] Erik Cambria et al. "New avenues in opinion mining and sentiment analysis". In: *IEEE Intelligent Systems* 33.2 (2018), pp. 15–23.

[3] Dorottya Demszky et al. "GoEmotions: A Dataset of Fine-Grained Emotions". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 2020, pp. 4048–4063.

[4] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Vol. 1. 2019, pp. 4171–4186.

[5] Andrea Esuli and Fabrizio Sebastiani. "SentiWordNet: A Lexical Resource for Sentiment Analysis". In: *Proceedings of LREC*. Vol. 6. 2006, pp. 417–422.

[6] Jing Fan et al. "Topic-Sentiment Analysis for Online Reviews Based on LDA and SVM". In: *Journal of Physics: Conference Series* 1683.1 (2020), p. 012078.

[7] Minqing Hu and Bing Liu. "Mining and Summarizing Customer Reviews". In: *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2004, pp. 168–177.

[8] C.J. Hutto and Eric E. Gilbert. "VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 8. 1. 2014, pp. 216–225.

[9] Thorsten Joachims. "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". In: *European Conference on Machine Learning*. 1998.

[10] Soo-Min Kim and Eduard H. Hovy. "Extracting Opinions with Conditional Random Fields". In: *Proceedings of the HLT-NAACL*. 2006.

[11] Bing Liu. *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers, 2012.

[12] Yinhan Liu et al. "RoBERTa: A Robustly Optimized BERT Pretraining Approach". In: *arXiv preprint arXiv:1907.11692* (2019).

[13] Saif M. Mohammad and Peter D. Turney. "Crowdsourcing a Word-Emotion Association Lexicon". In: *Computational Intelligence* 29.3 (2013), pp. 436–465.

[14] Finn Årup Nielsen. "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs". In: *Proceedings of the 1st Workshop on 'Affect in Text and Speech'*. Vol. 11. 1. 2011, pp. 9–14.

[15] Bo Pang and Lillian Lee. "Opinion Mining and Sentiment Analysis". In: *Foundations and Trends in Information Retrieval* 2.1–2 (2008), pp. 1–135.

[16] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. "Thumbs up? Sentiment Classification using Machine Learning Techniques". In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. 2002, pp. 79–86.

[17] Hyeongmin Park and Joonas Lee. "Impact of User Feedback on Sentiment Analysis: An Empirical Study of Movie Reviews". In: *Journal of Big Data* 5.1 (2018), pp. 1–19.

[18] Matthew Purver. "Location-Based Sentiment Analysis on Twitter: A Comparison of Methods". In: *Proceedings of the IEEE International Conference on Social Computing (SocialCom)*. 2011, pp. 192–199.

[19] Alec Radford et al. *Improving Language Understanding by Generative Pre-Training*. OpenAI blog. 2018.

[20] Gerard Salton and Christopher Buckley. "Term-weighting Approaches in Automatic Text Retrieval". In: *Information Processing & Management* 24.5 (1988), pp. 513–523.

[21] Andranik Tumasjan et al. "Predicting Elections with Twitter: What 140 Characters Reveal About Political Sentiment". In: *Proceedings of the International AAAI Conference on Web and Social Media*. Vol. 4. 1. 2010.

[22] Ashish Vaswani et al. "Attention Is All You Need". In: *Advances in Neural Information Processing Systems*. Vol. 30. 2017.

[23] Tom Young et al. "Recent Trends in Deep Learning for Natural Language Processing". In: *IEEE Computational Intelligence Magazine* 13.3 (2018), pp. 55–73.

.10 Code-appendix (with Jupyter notebook Python)

Listing 1: *REQUIRED INSTALLATIONS*

```
pip install -r requirements.txt
```

Listing 2: *IMPORT LIBRARIES*

```
import pandas as pd
import numpy as np
import ast
import re
import os
import emoji
import warnings
from collections import Counter
from difflib import SequenceMatcher

# Visualization
import seaborn as sns
import matplotlib.pyplot as plt

# NLP & Text
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

# Transformers
from transformers import pipeline

# Progress Bar
from tqdm.notebook import tqdm

# TF-IDF and Word2Vec
from sklearn.feature_extraction.text import TfidfVectorizer
from gensim.models import Word2Vec
```

Listing 3: *DOWNLOAD NLTK RESOURCES*

```
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('omw-1.4')
nltk.download('stopwords')

# Activate tqdm for pandas
tqdm.pandas()
```

Listing 4: *LOAD DATA*

```
comments = pd.read_csv("comments.csv")
posts = pd.read_csv("posts_answers.csv")
users = pd.read_csv("users.csv")

warnings.filterwarnings('ignore')
```

Listing 5: *INITIAL EXPLORATION (1/2)*

```
print("--- Initial Exploration: Comments Data ---")
print("\nHead of Comments:")
display(comments.head())

print("\nMissing Values:")
display(comments.isnull().sum())

print("\nDuplicate Rows:", comments.duplicated().sum())
print("Unique Users:", comments['user_id'].nunique())
print("Unique Posts:", comments['post_id'].nunique())

print("\n\n--- Initial Exploration: Posts_answers Data ---")
display(posts.head())
display(posts.isnull().sum())
print("Duplicate Rows:", posts.duplicated().sum())

plt.figure(figsize=(10, 5))
sns.histplot(posts['score'].dropna(), bins=50, kde=True, color='purple')
plt.title('Distribution of Post Scores')
plt.xlabel('Score')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()
```

Listing 6: *INITIAL EXPLORATION (2/2)*

```
print("\n\n--- Initial Exploration: Users Data ---")
display(users.head())
display(users.isnull().sum())
print("Duplicate Rows:", users.duplicated().sum())

plt.figure(figsize=(10, 5))
sns.histplot(users['reputation'].dropna(), bins=50, kde=True, color='green')
plt.title('Distribution of User Reputation')
plt.xlabel('Reputation')
plt.ylabel('Frequency')
plt.tight_layout()
plt.show()

print("\nTop 10 User Locations:")
display(users['location'].value_counts().head(10))
```

Listing 7: *TEXT CLEANING (1/3)*

```
# Regex Patterns
URL_PATTERN = re.compile(r"http\S+|www.\S+")
MEANINGFUL_TEXT_PATTERN = re.compile(r"\w{3,}")
EMOJI_PATTERN = re.compile("[\U0001F600-\U0001F64F"
                           "\U0001F300-\U0001F5FF"
                           "\U0001F680-\U0001F6FF"
                           "\U0001F1E0-\U0001F1FF]+", flags=re.UNICODE)
```

Listing 8: *TEXT CLEANING (2/3)*

```
def has_meaningful_text(text):
    text = URL_PATTERN.sub("", text)
    return bool(MEANINGFUL_TEXT_PATTERN.search(text))

def remove_emojis(text):
    return EMOJI_PATTERN.sub("", text)

def clean_text(text, lower=True, strip=True, remove_urls=True, remove_emojis_flag=True):
```

```
    if pd.isna(text):
        return ""
    if lower:
        text = text.lower()
    if remove_urls:
        text = URL_PATTERN.sub("", text)
    if remove_emojis_flag:
        text = remove_emojis(text)
    if strip:
        text = text.strip()
    return text
```

Listing 9: *TEXT CLEANING (3/3)*

```
def clean_comments(df, text_col="text", min_words=3, drop_duplicates=True, remove_emojis_flag=True, lower=True):
    df = df.copy()
    df = df[df[text_col].notna()]
    df = df[df[text_col].str.strip() != ""]
    df[text_col] = df[text_col].apply(lambda x: clean_text(x, lower=lower, remove_emojis_flag=remove_emojis_flag))
    if drop_duplicates:
        df = df.drop_duplicates(subset=[text_col])
    df = df[df[text_col].str.split().str.len() >= min_words]
    df = df[df[text_col].apply(has_meaningful_text)]
    return df.reset_index(drop=True)

# Apply cleaning
comments = clean_comments(comments, min_words=3, remove_emojis_flag=True)
```

Listing 10: *BASIC STATS VISUALS*

```
print("Number of comments:", len(comments))
print("Unique users:", comments["user_id"].nunique())
print("Missing values in text:", comments["text"].isna().sum())

# Word Count Distribution
comments["word_count"] = comments["text"].apply(lambda x: len(str(x).split()))

plt.figure(figsize=(10, 6))
sns.histplot(comments["word_count"], bins=30, kde=True, color="skyblue")
```

```
plt.title("Distribution of Comment Lengths (Word Count)")
plt.xlabel("Word Count")
plt.ylabel("Frequency")
plt.tight_layout()
plt.show()
```

Listing 11: *WORD FREQUENCY ANALYSIS*

```
stop_words = set(stopwords.words("english"))

def tokenize(text):
    return [word.lower() for word in str(text).split() if word.lower() not in stop_words and word.isalpha()]

all_words = sum(comments["text"].apply(tokenize).tolist(), [])
top_words = Counter(all_words).most_common(20)

# Plot Top Words
words, counts = zip(*top_words)
plt.figure(figsize=(10, 6))
ax = sns.barplot(x=list(counts), y=list(words), palette="viridis")
plt.title("Top 20 Most Frequent Words in Comments")
plt.xlabel("Frequency")
plt.ylabel("Word")

for i, count in enumerate(counts):
    ax.text(count + 0.5, i, str(count), va='center', fontsize=10)
plt.tight_layout()
plt.show()
```

Listing 12: *FEATURE EXTRACTION*

```
# Ensure text column is string type
comments['text'] = comments['text'].fillna('').astype(str)

# TF-IDF Vectorization
tfidf_vectorizer = TfidfVectorizer(max_features=28000, stop_words='english', ngram_range=(1, 1))
tfidf_matrix = tfidf_vectorizer.fit_transform(comments['text'])

print(f"TF-IDF Matrix Shape: {tfidf_matrix.shape}")
print(f"Unique Features Extracted: {len(tfidf_vectorizer.
```

```
    get_feature_names_out())}")
```

Listing 13: *TOKENIZATION*

```
lemmatizer = WordNetLemmatizer()

def tokenize_and_lemmatize(text):
    tokens = word_tokenize(text.lower())
    return [lemmatizer.lemmatize(word) for word in tokens if word.isalpha() and word not in stop_words]

comments['lemmatized_tokens'] = comments['text'].progress_apply(tokenize_and_lemmatize)

# Generate bigrams
bigrams_list = []
for tokens in comments['lemmatized_tokens']:
    bigrams_list.extend(list(nltk.bigrams(tokens)))

bigram_counts = Counter(bigrams_list)
top_bigrams = bigram_counts.most_common(20)
```

Listing 14: *BIGRAM PLOT*

```
# Plot Top Bigrams
plt.figure(figsize=(10, 7))
bigram_words = [" ".join(bg) for bg, count in top_bigrams]
bigram_counts_val = [count for bg, count in top_bigrams]

ax = sns.barplot(x=bigram_counts_val, y=bigram_words, palette="coolwarm")
plt.title("Top 20 Most Frequent Bigrams in Comments")
plt.xlabel("Frequency")
plt.ylabel("Bigram")

for i, count in enumerate(bigram_counts_val):
    ax.text(count + 0.5, i, str(count), va='center', fontsize=10)
plt.tight_layout()
plt.show()
```

Listing 15: *GoEmotions FROM HUGGING FACE*

```
import time
# Define batch inference function
def batch_infer(df, batch_size=16, model_name="SamLowe/roberta-base-go_emotions"):
    classifier = pipeline("text-classification", model=model_name, top_k=None)
    results = []
    texts = df["text"].tolist()

    for i in tqdm(range(0, len(texts), batch_size), desc="Batch Inference"):
        batch = [str(t)[:512] for t in texts[i:i + batch_size]]
        batch_result = classifier(batch)
        results.extend(batch_result)

    return results

# Run the function and time it
print("Running emotion detection with GoEmotions model...")

start_time = time.time()
comments["emotions"] = batch_infer(comments)
comments.to_csv("comments_with_emotions.csv", index=False)
print("---> Saved emotions per comment to comments_with_emotions.csv")

end_time = time.time()

print(f"---> Emotion inference completed in {end_time - start_time:.2f} seconds.")
```

Listing 16: *EMOTION DATA PROCESSING (1/3)*

```
# Load Comments with Emotions
comments = pd.read_csv("comments_with_emotions.csv")
comments["emotions"] = comments["emotions"].apply(ast.literal_eval)

print(f" ---> Loaded {len(comments)} comments with parsed emotions.")
```

Listing 17: *EMOTION DATA PROCESSING (2/3)*

```
# Extract Emotion Labels Above a Score Threshold
def extract_labels(e_list, threshold=0.3):
    """
    Filters emotion predictions above a given score threshold.
    """
    return [e["label"] for e in e_list if e["score"] > threshold]

comments["emotion_labels"] = comments["emotions"].apply(extract_labels)
print(" ---> Extracted emotion labels for each comment.")
```

Listing 18: *EMOTION DATA PROCESSING (3/3)*

```
# Explode Multi-Emotion Comments into Rows
comments_exp = comments.explode("emotion_labels")
comments_exp = comments_exp[comments_exp["emotion_labels"].notna()]
print(f" ---> Exploded to {len(comments_exp)} rows (one row per emotion per comment).")

# Ensure ID Columns Are Numeric for Merging
comments_exp["user_id"] = pd.to_numeric(comments_exp["user_id"], errors="coerce")
comments_exp["post_id"] = pd.to_numeric(comments_exp["post_id"], errors="coerce")
print(" ---> Converted user_id and post_id to numeric for merging.")
```

Listing 19: *MERGE COMMENTS WITH USERS AND POSTS*

```
# Merge Comments with Users and Posts
def merge_all(comments_df, users_df, posts_df):
    """
    Merges comment data with user and post metadata.
    """
    merged = comments_df.merge(users_df, left_on="user_id", right_on="id", how="left", suffixes=('', '_user'))
    merged = merged.merge(posts_df, left_on="post_id", right_on="id", how="left", suffixes=('', '_post'))
    return merged

df = merge_all(comments_exp, users, posts)
print(f" ---> Merged with users and posts. Final shape: {df.shape}")
```

Listing 20: *BIN NUMERIC COLUMNS INTO CATEGORIES*

```
def bin_column(df, col, bins, labels, new_col):
    """
    Adds a categorical column based on binning of a numeric column.
    """
    df[new_col] = pd.cut(df[col], bins=bins, labels=labels)
    return df
```

Listing 21: *REPUTATION GROUPING*

```
df = bin_column(
    df, "reputation",
    bins=[-1, 100, 1000, 5000, 20000, float("inf")],
    labels=["Very Low", "Low", "Medium", "High", "Very High"],
    new_col="rep_group"
)

# Post Score Grouping
df = bin_column(
    df, "score_post",
    bins=[-10, 0, 5, 20, 100, float("inf")],
    labels=["Negative", "Low", "Moderate", "High", "Very High"],
    new_col="post_score_group"
)

# Comment Score Grouping
df = bin_column(
    df, "score",
    bins=[-5, 0, 5, 20, 100, float("inf")],
    labels=["Negative", "Low", "Medium", "High", "Very High"],
    new_col="comment_score_group"
)

print(" ---> Created reputation and post score groups.")
```

Listing 22: *DISPLAY SAMPLE OUTPUT*

```
# Display Sample Output
display(df[[
    "text", "emotion_labels",
    "reputation", "rep_group",
    "score_post", "post_score_group"
```

```
]].sample(5))
```

Listing 23: *TOP-SCORING COMMENTS BY EMOTION*

```
top_comments_per_emotion = (
    df[df["emotion_labels"].notna()]
    .groupby("emotion_labels")
    .apply(lambda g: g.sort_values("score", ascending=False).head(1))
)[["text", "score", "display_name", "emotion_labels"]]

print("---> Top-Scoring Comments by Emotion:")
display(top_comments_per_emotion.reset_index(drop=True))
```

Listing 24: *MOST CONFIDENT EMOTION PREDICTIONS*

```
# Most Confident Emotion Predictions
df["emotion_confidence"] = df["emotions"].apply(lambda x: max([e["score"]↩
    for e in x]) if x else 0)
df_top_confident = df.sort_values("emotion_confidence", ascending=False).↩
    head(10)

print("---> Most Confident Emotion Predictions:")
display(df_top_confident[["text", "emotion_labels", "emotion_confidence",↩
    "score", "display_name"]])
```

Listing 25: *BAR PLOT - FREQUENCY OF ALL 27 EMOTIONS*

```
from collections import Counter
all_emotions = sum(comments["emotion_labels"].tolist(), [])
emotion_counts = pd.Series(Counter(all_emotions)).sort_values(ascending=↩
    False)

plt.figure(figsize=(12, 8))
ax = sns.barplot(x=emotion_counts.values, y=emotion_counts.index, palette↩
    ="coolwarm")
plt.title("Frequency of All 27 Emotions")
plt.xlabel("Count")
plt.ylabel("Emotion")
for i, (value, label) in enumerate(zip(emotion_counts.values, ↩
    emotion_counts.index)):
    ax.text(value + 2, i, str(value), va='center', fontsize=9)
plt.tight_layout()
```

```
plt.show()
```

Listing 26: *BAR PLOT - FREQUENCY OF ALL 27 EMOTIONS EXCLUDING NEUTRAL*

```
# Bar Plot: Emotions (Excluding Neutral)
non_neutral_emotions = [e for e in all_emotions if e != "neutral"]
non_neutral_counts = pd.Series(Counter(non_neutral_emotions)).sort_values(ascending=False)

plt.figure(figsize=(12, 8))
ax = sns.barplot(x=non_neutral_counts.values, y=non_neutral_counts.index, palette="coolwarm")
plt.title("Frequency of Emotions (Excluding Neutral)")
plt.xlabel("Count")
plt.ylabel("Emotion")
for i, (value, label) in enumerate(zip(non_neutral_counts.values, non_neutral_counts.index)):
    ax.text(value + 2, i, str(value), va='center', fontsize=9)
plt.tight_layout()
plt.show()
```

Listing 27: *EMOTION DISTRIBUTION BY COMMENT SCORE*

```
# Plot: Emotion Distribution by Comment Score
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x="comment_score_group", hue="emotion_labels", palette="tab20")
plt.title("Emotion Distribution by Comment Score (Popularity)")
plt.xlabel("Comment Score Group")
plt.ylabel("Count")
plt.legend(loc="upper right", bbox_to_anchor=(1.25, 1))
plt.tight_layout()
plt.show()
```

Listing 28: *EMOTION DISTRIBUTION BY COMMENT SCORE EXCLUDING NEUTRAL*

```
df_no_neutral = df[df["emotion_labels"] != "neutral"]

plt.figure(figsize=(12, 6))
sns.countplot(data=df_no_neutral, x="comment_score_group", hue="emotion_labels", palette="tab20")
plt.title("Emotion Distribution by Comment Score (Without Neutral)")
plt.xlabel("Comment Score Group")
plt.ylabel("Count")
plt.legend(loc="upper right", bbox_to_anchor=(1.25, 1))
plt.tight_layout()
plt.show()
```

Listing 29: *RE-BIN POST SCORES FOR PLOTTING*

```
# Re-bin Post Scores for Plotting
df["post_score_group"] = pd.cut(
    df["score_post"],
    bins=[-10, 0, 5, 20, 100, float("inf")],
    labels=["Negative", "Low", "Medium", "High", "Very High"]
)
```

Listing 30: *EMOTION DISTRIBUTION PLOT BY POST SCORE*

```
plt.figure(figsize=(12, 6))
sns.countplot(data=df, x="post_score_group", hue="emotion_labels", palette="tab20")
plt.title("Emotion Distribution by Post Score")
plt.xlabel("Post Score Group")
plt.ylabel("Emotion Count")
plt.legend(loc="upper right", bbox_to_anchor=(1.25, 1))
plt.tight_layout()
plt.show()
```

Listing 31: *EMOTION DISTRIBUTION PLOT BY POST SCORE EXCLUDING NEUTRAL*

```
# Exclude Neutral

plt.figure(figsize=(12, 6))
sns.countplot(data=df_no_neutral, x="post_score_group", hue="emotion_labels", palette="tab20")
plt.title("Emotion Distribution by Post Score (Without Neutral)")
plt.xlabel("Post Score Group")
plt.ylabel("Count")
plt.legend(loc="upper right", bbox_to_anchor=(1.25, 1))
plt.tight_layout()
plt.show()
```

Listing 32: *GEOGRAPHICAL EMOTION ANALYSIS*

```
df_clean = df.copy()
df_clean["location"] = df_clean["location"].replace(["United States", "USA"], "United States")

top_locations_clean = df_clean["location"].value_counts().nlargest(10).index
df_top_loc_clean = df_clean[df_clean["location"].isin(top_locations_clean)]

plt.figure(figsize=(12, 6))
sns.countplot(data=df_top_loc_clean, x="location", order=top_locations_clean)
plt.title("Top 10 User Locations by Number of Comments")
plt.xlabel("Location")
plt.ylabel("Number of Comments")
plt.xticks(rotation=45, ha="right")
plt.tight_layout()
plt.show()
```

Listing 33: *EMOTIONS HEATMAP BY LOCATION*

```
df["location"] = df["location"].replace({"USA": "United States", "U.S.": "United States"})
top_locations = df["location"].value_counts().nlargest(10).index
df_top_loc = df[df["location"].isin(top_locations)]
```

```
4
5  heatmap_data = df_top_loc.groupby(["emotion_labels", "location"]).size().↩
       unstack(fill_value=0)
6
7  plt.figure(figsize=(14, 8))
8  sns.heatmap(
9      heatmap_data,
10     annot=True,
11     fmt="d",
12     cmap="YlGnBu",
13     linewidths=0.5,
14     cbar_kws={"label": "Count"}
15 )
16 plt.title("Heatmap: Emotions by User Location (Top 10 Locations)")
17 plt.xlabel("Location")
18 plt.ylabel("Emotion")
19 plt.xticks(rotation=45, ha="right")
20 plt.tight_layout()
21 plt.show()
```

Listing 34: *EMOTIONS HEATMAP BY LOCATION EXCLUDING NEUTRAL*

```
1  # Heatmap: Emotions by Location (Excluding Neutral)
2  heatmap_no_neutral = df_top_loc_clean[df_top_loc_clean["emotion_labels"] ↩
       != "neutral"] \
3      .groupby(["emotion_labels", "location"]).size().unstack(fill_value=0)
4
5  plt.figure(figsize=(14, 8))
6  sns.heatmap(
7      heatmap_no_neutral,
8      annot=True,
9      fmt="d",
10     cmap="YlGnBu",
11     linewidths=0.5,
12     cbar_kws={"label": "Count"}
13 )
14 plt.title("Heatmap: Emotions by User Location (Excluding Neutral)")
15 plt.xlabel("Location")
16 plt.ylabel("Emotion")
17 plt.xticks(rotation=45, ha="right")
18 plt.tight_layout()
19 plt.show()
```

Listing 35: *UPVOTES BOX-PLOT BY EMOTIONS*

```
# Box Plot: Upvotes by Emotion
plt.figure(figsize=(14, 8))
sns.boxplot(
    data=df,
    y="emotion_labels",
    x="up_votes",
    palette="coolwarm",
    showfliers=False,
    width=0.6
)
plt.title("Upvotes by Emotion", fontsize=14, pad=15)
plt.xlabel("Upvotes", fontsize=12)
plt.ylabel("Emotion", fontsize=12)
plt.ticklabel_format(style="plain", axis="x")
sns.despine(trim=True)
plt.grid(axis="x", linestyle="--", alpha=0.4)
plt.tight_layout()
plt.show()
```

Listing 36: *DOWNVOTES BOX-PLOT BY EMOTIONS*

```
# Box Plot: Downvotes by Emotion
plt.figure(figsize=(14, 8))
sns.boxplot(
    data=df,
    y="emotion_labels",
    x="down_votes",
    palette="coolwarm",
    showfliers=False
)
plt.title("Downvotes by Emotion", fontsize=14, pad=15)
plt.xlabel("Downvotes", fontsize=12)
plt.ylabel("Emotion", fontsize=12)
plt.tight_layout()
plt.show()
```