Real or Not? NLP with Disaster Tweets

Γεώργιος Χατζηαντώνης[1], Ανδρέας Τσουλούπας[2], Κωνσταντίνος Δημητρίου [3]
Department of Computer Science
University of Cyprus
Email: [1]ghadji02@cs.ucy.ac.cy, [2]atsoul03@cs.ucy.ac.cy, [3]kdemet06@cs.ucy.ac.cy

***Περίληψη*—Στις μέρες μας το Twitter έχει γίνει ένα από τα σημαντικότερα κανάλια επικοινωνίας σε καταστάσεις έκτακτης ανάγκης. Η συνεχής παρουσία των έξυπνων κινητών στη ζωή των ανθρώπων, τους επιτρέπει να ανακοινώσουν μια κατάσταση έκτακτης ανάγκης που παρατηρούν σε πραγματικό χρόνο. Εξαιτίας αυτού όλο και περισσότεροι οργανισμοί όπως πρακτορεία ειδήσεων ή οργανώσεις αρωγής καταστροφών ενδιαφέρονται να παρακολουθούν το Twitter. Στόχος της συγκεκριμένης εργασίας είναι να προβλέψουμε αποδοτικά κατά πόσο ένα Tweet αναφέρεται σε κάποια καταστροφή ή όχι**

Λέξεις Κλειδιά*—*Καταστροφή, Tweet, Προεπεξεργασία, Διανυσματοποιητές, Μηχανική Μάθηση, Cross validation, Grid search, Train set, Test set, F1-weighted score.

I. Εισαγωγη

Η εργασία αυτή αφορούσε τη διαχείριση του περιεχομένου των tweets, επομένως ακολουθήθηκε η διαδικασία της επεξεργασίας της φυσικής γλώσσας (NLP).

Αρχικά θα γίνει αναφορά στη μελέτη που κάναμε στα δεδομένα τα οποία είχαμε στη διάθεση μας, σχηματίζοντας κάποιες γραφικές για καλύτερη κατανόηση του περιεχομένου τους. Ακολούθως θα περιγραφεί η προεπεξεργασία που έγινε στο περιεχόμενο των tweets. Στη συνέχεια θα γίνει σύντομη αναφορά στους διανυσματοποιητές που μελετήθηκαν, οι οποίοι σκοπό είχαν την μετατροπή του περιεχομένου των tweets σε διανύσματα αριθμών έτσι ώστε να μπορούν να δοθούν στους αλγόριθμους μηχανικής μάθησης. Τέλος θα περιγράψουμε τη μεθοδολογία που ακολουθήσαμε στα πειράματα μας και θα παρουσιάσουμε τα αποτελέσματα που πήραμε. Αυτά τα αποτελέσματα καθόρισαν και το τελικό μοντέλο, δηλαδή τη μέθοδο προεπεξεργασίας, τη μέθοδο διανυσματοποίησης και τον αλγόριθμο μηχανικής μάθησης μαζί με τις καλύτερες του παραμέτρους.

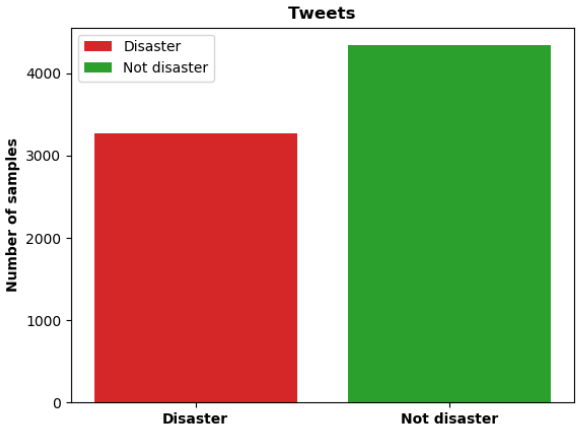
II. Μελετη Δεδομενων

Το σύνολο δεδομένων που είχαμε στη διάθεση μας αποτελούνταν από 7613 tweets. Για κάθε tweet είχαμε τις εξής πληροφορίες: id ένας μοναδικός ακέραιος αριθμός που προσδιορίζει κάθε tweet, text το περιεχόμενο του tweet, location η τοποθεσία από την οποία το tweet στάλθηκε, keyword μια συγκεκριμένη λέξη που χαρακτηρίζει το tweet και target προσδιορίζει εάν το tweet αναφέρεται σε μία καταστροφή (1) ή όχι (0).

Πριν προχωρήσουμε στην επεξεργασία των δεδομένων μελετήσαμε τη μορφή τους έτσι ώστε να είμαστε βέβαιοι για τις μεθόδους που θα χρησιμοποιήσουμε. Δηλαδή έχοντας μια καλύτερη γνώση για τα υπάρχον δεδομένα θα κατευθύναμε τη μελέτη μας προς κάποια κατεύθυνση.

A. Γεννικές πληροφορίες για τα δεδομένα

Τα δεδομένα αποτελούνταν από 7613 tweets εκ των οποίων τα 3271 αναφέρονταν σε πραγματικές καταστροφές και τα υπόλοιπα 4342 δεν σχετίζονταν με κάποια καταστροφή. Άρα τα δεδομένα μας ήταν σχεδόν ισοζυγισμένα και δε θα είχαμε πρόβλημα κατά την εξάσκηση αλγορίθμων μηχανικής μάθησης. Στην εικόνα 1 βλέπουμε αυτή την κατανομή.

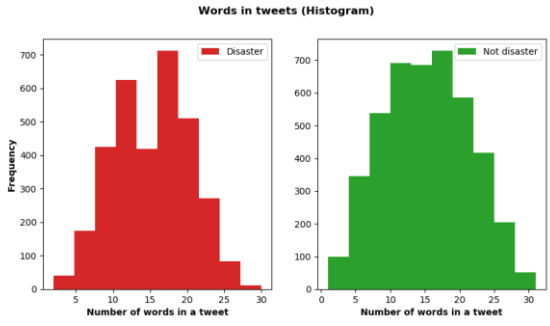


Εικόνα 1. Tweets που αναφέρονται σε καταστροφές και όχι.

Επίσης με μια πρώτη ματιά ανοίγοντας το αρχείο των δεδομένων, διακρίναμε πως το περιεχόμενο ήταν στην Αγγλική γλώσσα.

B. Πλήθος λέξεων σε κάθε tweet

Στην εικόνα 2 παρουσιάζεται το ιστόγραμμα που περιέχει τον αριθμό των λέξεων σε κάθε tweet. Υπολογίζοντας βρήκαμε πως ο μέσος όρος του αριθμού των λέξεων είναι 15 λέξεις στα tweets που αναφέρονται σε καταστροφές και 14 λέξεις σε αυτά που δεν αναφέρονται σε καταστροφές. Λόγω του ότι ο μέσος όρος του αριθμού των λέξεων στα tweets είναι σχετικά μικρός αποφασίσαμε στους διανυσματοποιητές TF-IDF και Counter Vectorizer (γνωστός και ως Bag of words) να μην εξετάσουμε n-grams με μέγεθος μεγαλύτερο από 2 λέξεις.

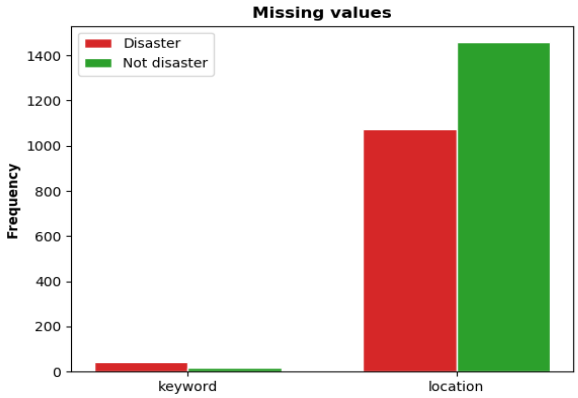


Εικόνα 2. Πλήθος λέξεων στα tweets.

C. Λέξεις κλειδιά και τοποθεσίες

Από το σύνολο δεδομένων που είχαμε παρατηρήσαμε ότι έλειπαν κάποιες τιμές από τις στήλες που αφορούσαν τις

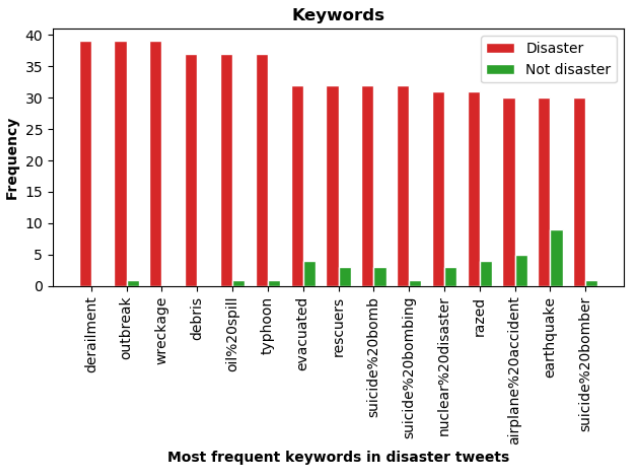
λέξεις κλειδιά και τις τοποθεσίες. Στην εικόνα 3 μπορούμε να δούμε σε πόσα tweets απουσιάζει αυτή η πληροφορία.



Εικόνα 3. Απουσια τιμών στις λέξεις κλειδιά και στις τοποθεσίες.

Κάνοντας τους υπολογισμούς είδαμε πως απουσιάζει περίπου το 0.8% των λέξεων κλειδιών και το 33% των τοποθεσιών. Αρχικά σκεφτήκαμε να μη λάβουμε υπόψιν καθόλου τις τοποθεσίες, αλλά καταλήξαμε στην ακόλουθη απόφαση. Όταν θα βρίσκαμε το καλύτερο μοντέλο χρησιμοποιώντας μόνο το περιεχόμενο του tweet, θα εξετάζαμε το κατά πόσο θα βοηθούσε αν κολλούσαμε στην αρχή του tweet τις λέξεις κλειδιά ή τις τοποθεσίες ή και τα δύο. Τα αποτελέσματα θα παρουσιαστούν στη συνέχεια.

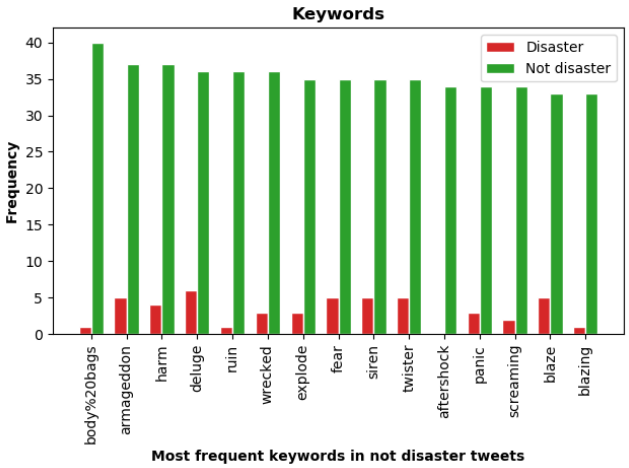
Ακόμη μια σημαντική παρατήρηση η οποία αφορούσε τις λέξεις κλειδιά είναι πως κάποιες από αυτές αφορούσαν σχεδόν πάντα είτε καταστροφές είτε όχι καταστροφές. Έτσι θεωρήσαμε αναγκαίο να εξετάσουμε την προσθήκη τις λέξεις κλειδιού στο διανυσματοποιητή. Αυτό όμως ίσως περιέχει κάποιο κίνδυνο αφού tweets που είναι εξαίρεση στον κανόνα θα προβλέπονταν λανθασμένα και έτσι θα αυξανόταν το σφάλμα μας. Κάποιες από τις ποιο συχνές λέξεις κλειδιά στα tweets με καταστροφές φαίνονται στην εικόνα 4 και για όχι καταστροφές στην εικόνα 5.



Εικόνα 4. Πιο συχνές λέξεις κλειδία στα tweets με καταστροφές.

D. Άλλα σχόλια

Έγιναν και άλλες προσπάθειες για άντληση πληροφοριών μόνο από τις γραφικές αλλά δεν είχαμε κάποια άλλη σημαντική παρατήρηση. Στο παράρτημα Α παρουσιάζονται κάποιες επιπλέων γραφικές που κατασκευάσαμε για την μελέτη των δεδομένων.



Εικόνα 5. Πιο συχνές λέξεις κλειδία στα tweets που δεν είναι καταστροφή.

III. Προεπεξεργασια

Ένα πολύ σημαντικό μέρος της εργασίας αφιερώθηκε στον τρόπο προεπεξεργασίας των δεδομένων, έτσι ώστε όταν θα δοθούν στους διανυσματοποιητές να παράξουν καλύτερα και πιο αντιπροσωπευτικά διανύσματα τιμών για κάθε tweet.

Αναλυτική αναφορά στα στάδια προεπεξεργασίας δεδομένων που εφαρμόστηκαν σε κάθε περίπτωση βρίσκεται στο Παράρτημα Β

A. Προεπεξεργασία κειμένου

1. Γενική προεπεξεργασία

Μέρος των tweets περιείχαν υπερσύνδεσμους σε άλλες ιστοσελίδες. Έγινε δοκιμή αντικαθιστώντας τον υπερσύνδεσμο με την λέξη ‘url’, αλλά βλέποντας σχετική μείωση στην επίδοση των μοντέλων ο υπερσύνδεσμος διαγραφόταν εντελώς. Με σκοπό να χρησιμοποιήσουμε το περιεχόμενο των σελίδων που έδειχναν οι υπερσύνδεσμοι, έγινε χρήση της βιβλιοθήκης BeautifulSoup[1], ώστε να εξάγουμε το κείμενο από την σελίδα. Ακολουθήθηκε διαγραφή των html tags, λέξεων/χαρακτήρων και μετά παρόμοια προεπεξεργασία του κειμένου από την σελίδα, όπως και το κείμενο των tweets. Παρ ’όλα αυτά φάνηκε ότι μεγάλο μέρος των υπερσυνδέσμων κατέληγαν σε σελίδες που δεν υπήρχαν, είτε δεν ήταν προσβάσιμες. Λαμβάνοντας επιπλέον υπόψη το γεγονός ότι το περιεχόμενο των σελίδων ήταν μεγάλου μεγέθους, και ότι εμπεριείχε άσχετες πληροφορίες, αποφασίστηκε να μην χρησιμοποιήσουμε καθόλου το κείμενο των ιστοσελίδων στη φάση των πειραμάτων.

Ακολούθησε αντικατάσταση χαρακτήρων html με unicode χαρακτήρες (π.χ. >, >, >). Επιπρόσθετα διαγράφηκαν όλοι οι non ascii και non printable χαρακτήρες, σημεία στίξης και αριθμοί. Θεωρήθηκε ότι η ύπαρξη των usernames (λέξεις που αρχίζουν με @) στο κείμενο δεν θα βοηθούσε στην κατηγοριοποίηση, έτσι διαγράφονταν. Τα emojis και emojicons με χρήση λεξικού αντικαθίστανται με αντίστοιχο κείμενο. Επαναλαμβανόμενοι χαρακτήρες πέραν των 2 συνεχόμενων, αντικαταστάθηκαν με 2, έτσι ώστε να βοηθηθεί ο αλγόριθμος διόρθωσης ορθογραφικών που θα τρέξει στην συνέχεια. Όλοι οι χαρακτήρες μετατράπηκαν σε lower case, και σε περίπτωση όπου υπήρχαν λέξεις υπό μορφή

camelCase, θεωρήθηκε καλό, μετά από παρατήρηση δείγματος, να χωριστούν στο σημείο που ξεκινά με κεφαλαίο. Τέτοια μορφή παρατηρήθηκε κυρίως στα hashtags. Με χρήση της βιβλιοθήκης Ekphrasis[2], hashtags που εμφανίζονταν στο κείμενο (λέξεις που αρχίζουν με #) διαχωρίστηκαν, άσχετος αν δεν περιείχαν κεφαλαία. Συντομογραφίες (abbreviations και contractions) αντικαταστάθηκαν χρησιμοποιώντας λεξικό. Τέλος αφαιρέθηκαν stop-words και εφαρμόστηκε lemmatization/stemming.

Υλοποίηση των όσων αναφέρθηκαν έγινε κυρίως με χρήση λεξικών και regular expressions. Να σημειωθεί ότι δεν εφαρμόστηκαν τα πιο πάνω βήματα προεπεξεργασίας σε όλα τα πειράματα

2. Προεπεξεργασία για χρήση σε μοντέλα τύπου BERT

Το Bidirectional Encoder Representations from Transformers (BERT)[3] είναι μία τεχνική εξεργασίας φυσικής γλώσσας με την χρήση προ-εκπαιδευμένων μοντέλων, η οποία υλοποιήθηκε από την Google και χρησιμοποιείται σε state-of-the-art συστήματα. Τα συγκεκριμένα μοντέλα χρησιμοποιήθηκαν αργότερα κατά τη φάση της διανυσματοποίησης του κειμένου. Με σκοπό την επίτευξη καλύτερου αποτελέσματος εφαρμόστηκε προεπεξεργασία στα δεδομένα που προορίζονταν για την συγκεκριμένη τεχνική, τέτοια ώστε το λεξικό που θα λάβουμε να πλησιάζει όσο το δυνατό πιο κοντά στα embeddings που χρησιμοποιήθηκαν στην φάση της προ-εκπαίδευσης των μοντέλων. Μετά από μελέτη φάνηκε ότι οι αριθμοί και τα url θα έπρεπε να διαγραφούν, αφού εμφανιζόταν πολλές φορές στα δεδομένα. Επιπλέον δεν εφαρμόστηκε lemmatization αφού δε θέλαμε να κάνουμε 2 διαφορετικές λέξεις τις ίδιες ούτε stemming, αφού οι λέξεις που προκύπτουν δεν υπήρχαν στο λεξικό που χρησιμοποιήθηκε στην προ-εκπαίδευση. Τέλος τα stop-words και τα σημεία στίξης δεν αφαιρέθηκαν αφού υπήρχαν στο λεξικό προ-εκπαίδευσης και θα "βοηθούσαν" το μοντέλο στην καλύτερη κατανόηση και διανυσματοποίηση των δεδομένων. Πιο συγκεκριμένα μετά από την επεξεργασία 70.81% των μοναδικών λέξεων που υπήρχαν στα tweets (συνολικά το 92.91% όλου του κειμένου), υπήρχε και στα embeddings. Να σημειωθεί ότι πριν την προεπεξεργασία μόνο το 28.03% των μοναδικών λέξεων (συνολικά το 71.99% όλου του κειμένου) υπήρχε και στα embeddings.

3. Προεπεξεργασία με χρήση της βιβλιοθήκης Ekphrasis

Για σκοπούς σύγκρισης της προεπεξεργασίας που υλοποιήθηκε με έτοιμες υπάρχουσες βιβλιοθήκες, επιλέγηκε η βιβλιοθήκη Ekphrasis. Η βιβλιοθήκη Ekphrasis αποτελεί εργαλείο επεξεργασίας κειμένου, προσανατολισμένο προς κείμενο από κοινωνικά δίκτυα, γι' αυτό θεωρήθηκε και κατάλληλο στην δική μας περίπτωση. Με την χρήση της συγκεκριμένης βιβλιοθήκης έγιναν tokenization, κανονικοποίηση λέξεων, τμηματοποίηση λέξεων (για διαχωρισμό hashtags) και διόρθωση ορθογραφίας, χρησιμοποιώντας στατιστικά λέξεων από twitter – 330 εκατομμύρια αγγλικά tweets. Στην φάση της κανονικοποίησης τα ακόλουθα στοιχεία αντικαταστάθηκαν με τα αντίστοιχα tags: url, email, percent, money, phone, user, time, date και number. Πέραν από την προεπεξεργασία που παρέχει η βιβλιοθήκη προχωρήσαμε αφαιρώντας τα σημεία στίξης, tags που παράγονται από την κανονικοποίηση, stop-words και εφαρμόζοντας lemmatazation ή stemming.

B. Προεπεξεργασία λέξεων κλειδιών

Όσον αφορά την στήλη 'keyword', δεν χρειάστηκε να γίνει ιδιαίτερη προεπεξεργασία - αντικαταστάθηκε ο χαρακτήρας %20 με κενό.

C. Προεπεξεργασία τοποθεσιών

Τα δεδομένα τοποθεσίας των tweets παρουσίαζαν μεγάλη διακύμανση και ποικιλία. Αρχικά αφαιρέθηκαν urls σε ιστοσελίδες, usernames, non ascii και non printable χαρακτήρες. Επίσης διαγράφηκαν σημεία στίξης, χαρακτήρες και tags HTML και όλοι οι χαρακτήρες μετατράπηκαν σε lowercase. Παρατηρήθηκε χρήση συντομογραφιών για πόλεις και χώρες. Έτσι δημιουργήθηκε λεξικό με τις συντομογραφίες των πολιτειών των ΗΠΑ, τα οποία εμφανίζονταν και τις περισσότερες φορές. Επίσης προστέθηκαν συχνές συντομογραφίες χωρών. Έγινε προσπάθεια ώστε να έχουν μια πιο συνεπή μορφή σε κύριες περιπτώσεις που παρατηρήθηκαν στην ανάλυση δεδομένων (π.χ. να εμφανίζεται πόλη/πολιτεία και χώρα) και να εξαλειφθούν επαναλαμβανόμενες λέξεις. Επίσης έγιναν συγκεκριμένες διορθώσεις σε συχνές εμφανίσεις. Η μορφή των δεδομένων και η κατάσταση στην οποία βρίσκονταν έκανε την προεπεξεργασία δύσκολη και δεν επιτεύχθηκε μια συνολική συνέπεια.

D. Διαγραφή διπλότυπων

Παρατηρήθηκε ότι μέρος των tweets είχαν το ίδιο κείμενο και είτε περιείχαν ίδιους ή διαφορετικούς υπερσυνδέσμους. Πιο συγκεκριμένα βρέθηκαν 354 πλειάδες tweets με ίδιο περιεχόμενο (συμπεριλαμβανομένου και με διαφορετικού url), εκ των οποίων οι 85 είχαν διαφορετικό target. Τα διπλότυπα tweets διαγράφηκαν και η τιμή του target ορίστηκε ως η τιμή που εμφανιζόταν περισσότερες φορές στα tweets με το ίδιο κείμενο. Στην περίπτωση που υπήρχε ίσος αριθμός, επιλεγόταν η τιμή 1. Συνολικά από την συγκεκριμένη επεξεργασία διαγράφηκαν 804 tweets.

IV. ΔΗΜΙΟΥΡΓΙΑ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ (ΔΙΑΝΥΣΜΑΤΟΠΟΙΗΣΗ)

Η δημιουργία χαρακτηριστικών είναι μια αναγκαία και πολύ σημαντική διαδικασία κατά την οποία το περιεχόμενο ενός κειμένου μετατρέπεται σε χαρακτηριστικά από αριθμούς. Δηλαδή κάθε κείμενο, που στη δική μας περίπτωση αφορούσε tweets, αντιπροσωπεύεται από ένα διάνυσμα με αριθμούς, οι διαστάσεις του οποίου καθορίζονται ανάλογα με τη μέθοδο διανυσματοποίησης που χρησιμοποιείται. Έτσι, φέρνοντας πλέον το κείμενο σε μια μορφή την οποία μπορούν να διαχειριστούν οι αλγόριθμοι μηχανικής μάθησης είμαστε σε θέση να τους εξασκήσουμε και στη συνέχεια με την ίδια μετατροπή σε νέα εισερχόμενα κείμενα να κάνουμε προβλέψεις.

Σε αυτό το σημείο είναι σημαντικό να αναφερθεί πως κάθε tweet είχε τρεις στήλες με κείμενο, μια με τη λέξη κλειδί, μια με την τοποθεσία και μια με το περιεχόμενο του tweet. Όπως θα δούμε στις πειραματικές μας μετρήσεις οι δύο στήλες της λέξης κλειδιού και της τοποθεσίας αξιοποιήθηκαν βάζοντας τες στην αρχή του περιεχόμενου του tweet στην τελευταία φάση πειραμάτων. Μελετήθηκαν όλοι οι συνδυασμοί, περισσότερες πληροφορίες θα αναφερθούν στη συνέχεια.

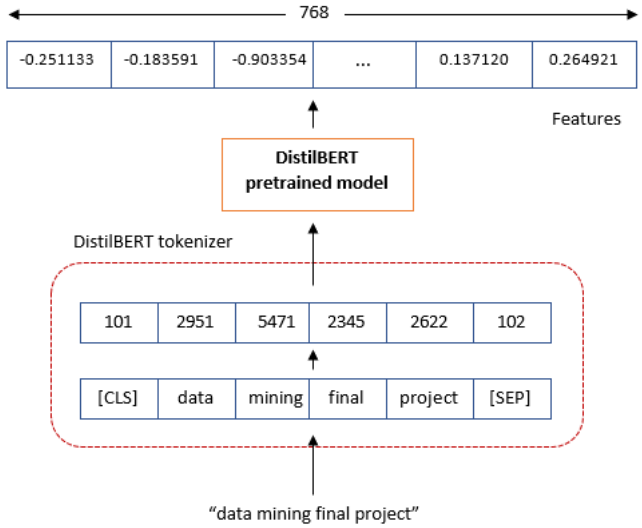
Στην εργασία αυτή όπως θα σας παρουσιάσουμε αμέσως μετά επικεντρωθήκαμε σε τέσσερεις διαφορετικές μεθόδους διανυσματοποίησης, DistilBERT, gensim doc2vec, TF-IDF

και Bag of words , δίνοντας κυρίως έμφαση στην DistilBERT.

A. Βασικός Διανυσματοποιητής

Η βασική μέθοδος που χρησιμοποιήσαμε για την διανυσματοποίηση ενός tweet, εμπίπτει στην κατηγορία των μετατροπέων του pytorch, γνωστοί ως transformers ή αλλιώς pytorch-pretrained-bert, οι οποίοι είναι κατάλληλοι για την κατανόηση της φυσικής γλώσσας. Λόγω έλλειψης υπολογιστικών πόρων χρησιμοποιήσαμε μια παραλλαγή του pretrained BERT μοντέλου, το οποίο ονομάζεται DistilBERT[4]. Η επιλογή μας βασίστηκε στο γεγονός πως το μοντέλο αυτό είναι εξασκημένο πάνω σε ένα τεράστιο όγκο κειμένων στην Αγγλική γλώσσα, το οποίο μας αφορούσε αφού τα tweets που είχαμε στη διάθεση μας ήταν στα Αγγλικά. Επιπλέον, το μοντέλο συγκρίνοντας το με το μοντέλο BERT χρησιμοποιεί 40% λιγότερες παραμέτρους και τρέχει σε 60% γρηγορότερο χρόνο διατηρώντας παράλληλα το 95% της απόδοσης του BERT.

Κάποια σημεία τα οποία έπρεπε να προσέξουμε χρησιμοποιώντας αυτόν τον διανυσματοποιητή είναι πως στα κείμενα στα οποία έγινε η εξάσκηση, το περιεχόμενο τους είχε μετατραπεί σε μικρούς χαρακτήρες. Επίσης, σημεία στίξης όπως τελείες, θαυμαστικά και κόμματα δεν είχαν αφαιρεθεί, συνεπώς διατήρηση τους στα tweets θα μας έδινε περισσότερη πληροφορία από την αφαίρεση τους. Το ίδιο ισχύει και για τις λέξεις stop words.



Εικόνα 6. Διαδικασία μετατροπής ενός κειμένου σε διάνυσμα αριθμών με τη χρήση του DistilBERT.

Πως όμως λειτουργεί αυτό το μοντέλο διανυσματοποίησης και πως μας φάνηκε χρήσιμο για την παραγωγή χαρακτηριστικών; Η απάντηση σε αυτό το ερώτημα είναι πολύ απλή. Το μοντέλο αυτό αφού το φορτώναμε, χρησιμοποιούσαμε τον tokenizer που έρχεται μαζί με το μοντέλο για να μετατρέψουμε κάθε tweet σε ακολουθία από tokens τα οποία είναι σε θέση να κατανοήσει το εξασκημένο μοντέλο. Είσοδος του μοντέλου ήταν όλες οι ακολουθίες από tokens, μια για κάθε tweet, και έξοδος, το αποτέλεσμα από κάθε μια από τις κρυμμένες καταστάσεις του μοντέλου. Εμάς μας αφορούσε το αποτέλεσμα της τελευταίας κρυμμένης κατάστασης και ποιο συγκεκριμένα το αποτέλεσμα του πρώτου token κάθε tweet. Το αποτέλεσμα αυτό έχει διαστάσεις 768 αριθμών και είναι το token το οποίο αφορά την κατηγοριοποίηση. Αυτό το διάνυσμα 768 αριθμών για κάθε tweet είναι και το παραγόμενο διάνυσμα που αντιπροσωπεύει το κάθε tweet και θα χρησιμοποιηθεί στην συνέχεια ως είσοδος στους αλγόριθμους μηχανικής μάθησης. Με αυτό τον τρόπο εκμεταλλευτήκαμε τη δύναμη που δίνει ένα εξασκημένο μοντέλο για την παραγωγή των γνωρισμάτων των tweets. Στην εικόνα 6 μπορείτε να δείτε πως γίνεται όλη η διαδικασία που περιγράψαμε πιο πάνω.

B. Άλλοι Διανυσματοποιητές

Όπως έχουμε προαναφέρει κύρια μας ενασχόληση ήταν να βελτιώσουμε τα αποτελέσματα με τον πιο πάνω διανυσματοποιητή, αλλά στα πειράματα μας δοκιμάσαμε και τις ακόλουθες μεθόδους διανυσματοποίησης.

Η μέθοδος TF-IDF είναι μια απλή μέθοδος. Ο σχηματισμός του διανύσματος που αναπαριστά το κάθε κείμενο γίνεται με βάση τον αριθμό των φορών που μια λέξη εμφανίζεται σε ένα αρχείο με τον αριθμό τον αρχείων στα οποία εμφανίζεται μια λέξη.

Από την άλλη, η μέθοδος Bag of words μετρά τον αριθμό των εμφανίσεων κάθε λέξης του λεξικού μέσα σε κάθε κείμενο και με αυτό τον τρόπο δημιουργεί το διάνυσμα των χαρακτηριστικών.

Τέλος, δοκιμάσαμε και τη μέθοδο που προσφέρεται στη βιβλιοθήκη gensim, την doc2vec, η οποία με παρόμοιο τρόπο όπως το DistilBERT παράγει για κάθε πρόταση ένα διάνυσμα τιμών. Η μέθοδος αυτή εγκαταλείφθηκε γρήγορα αφού δεν καταφέραμε να βρούμε ένα καλό εξασκημένο μοντέλο και η εξάσκηση του γινόταν με τα tweets που είχαμε το οποίο δεν έδιναν καλό αποτέλεσμα αφού ήταν σχετικά λίγα.

V. Μεθοδολογια Πειραματων

Κατά τις πειραματικές μας μετρήσεις προσπαθήσαμε να βρούμε τον συνδυασμό που θα μας έδινε το καλύτερο σκορ. Όταν αναφερόμαστε σε συνδυασμό εννοούμε τρία πράγματα, ένα την μέθοδο προεπεξεργασίας στα tweets, δύο το καλύτερο μοντέλο διανυσματοποίησης και τρία τον καλύτερο αλγόριθμο μηχανικής μάθησης μαζί με τις καλύτερες παραμέτρους του. Οι αλγόριθμοι μηχανικής μάθησης που χρησιμοποιήθηκαν ήταν οι Logistic Regression, K-nearest neighbors, SVC, Random Forest και Decision Trees από την βιβλιοθήκη sklearn[5].

Σημαντική Σημείωση – Σε κάθε περίπτωση που εξάταζαμε κάναμε 10 – folds cross validation στο δικό μας train set και στη συνέχεια ελέγχαμε το f1-weighted score στο δικό μας test set. Έτσι από το cross validation μπορούσαμε να δούμε αν κάναμε overfitting ή κατά πόσο υπήρχε μεγάλη τυπική απόκλιση. Συνεπώς μπορούσαμε να είμασταν βέβαιοι για το αποτέλεσμα αν είχαμε μικρή απόκλιση και κοντινό αποτέλεσμα στο test set από αυτό που πήραμε από το cross validation.

Η μεθοδολογία που ακολουθήσαμε στα πειράματα μας ακολούθησε τα πιο κάτω βήματα.

A. Διαχωρισμός του αρχικού dataset σε train και test

Από τον διαγωνισμό είχαμε στη διάθεση μας δύο datasets, το ένα από αυτά ήταν χωρίς την κατηγορία των tweets, επομένως για να μπορέσουμε να αξιολογήσουμε τους διάφορους συνδυασμού που κατασκευάζαμε έπρεπε να χωρίσουμε σε δικό μας train και test. Αυτό έγινε μια φορά στην αρχή και ο διαχωρισμός ήταν 80% train – 20% test. Όλες οι αξιολογήσεις έγιναν με βάση αυτά τα datasets.

B. *Εύρεση καλύτερου ζεύγους διανυσματοποιητή και αλγορίθμου μηχανικής μάθησης για κάθε μέθοδο προεπεξεργασίας*

Σε αυτό το βήμα εξετάσαμε για 8 διαφορετικούς τρόπους προεπεξεργασίας των tweets και για κάθε ένα από αυτούς τους τρόπους βρήκαμε το καλύτερο ζεύγος μεθόδου διανυσματοποίησης και αλγόριθμου μηχανικής μάθησης από ένα εύρος 12 διαφορετικών διανυσματοποιητών (κυρίως δοκιμάζαμε για παραμέτρους στους TF-IDF και Bag of words). Στους αλγόριθμους μηχανικής μάθησης κρατήσαμε τις default παραμέτρους. Το καλύτερο ζεύγος ήταν αυτό με το μεγαλύτερο σκορ στο test set. Δες σημείωση.

C. *Grid search στους 4 καλύτερους συνδυασμούς του βήματος B*

Για κάθε τρόπο προεπεξεργασίας είχαμε μείνει με ένα ζεύγος διανυσματοποιητή μαζί με τον αλγόριθμο μηχανικής μάθησης (πιο συγκεκριμένα αφήσαμε 2 αλγόριθμους μηχανικής μάθησης όπως θα δούμε στη συνέχεια). Επιλέξαμε τους τέσσερεις πιο υποσχόμενους συνδυασμούς και εφαρμόσαμε 10-folds cross validation grid search (sklearn) στον αλγόριθμο μηχανικής μάθησης του ζεύγους. Δηλαδή ελέγχαμε ένα μεγάλο εύρος τιμών των παραμέτρων των αλγορίθμων μηχανικής μάθησης, με την ελπίδα πως θα βρίσκαμε τις καλύτερες παραμέτρους τους. Όταν το grid search μας επέστρεφε τις καλύτερες παραμέτρους κάναμε τη διαδικασία που περιγράφεται στη σημείωση.

D. *Grid search στο καλύτερο αποτέλεσμα του βήματος C*

Μετά την ολοκλήρωση του βήματος C επιλέξαμε τον καλύτερο συνδυασμό. Εφαρμόσαμε για ακόμη μια φορά grid search αλλά αυτή την φορά οι παράμετροι που ελέγξαμε ήταν πιο κοντά στις καλύτερες παραμέτρους από το βήμα C για τον συγκεκριμένο συνδυασμό (π.χ. αν είχαμε την παράμετρο X με καλύτερο αποτέλεσμα από το βήμα C να είναι 10, τότε ελέγχαμε για X ίσο με 7,8,9,10 κλπ.) . Έτσι σε αυτό το βήμα προσπαθήσαμε να βελτιώσουμε την απόδοση του μοντέλου μας ψάχνοντας περισσότερες παραμέτρους, αλλά κατευθυνόμενοι από το προηγούμενο βήμα. Στο τέλος ίσως να παίρναμε καλύτερες παραμέτρους και ακολούθως ξαναεφαρμόζαμε τη διαδικασία της σημείωσης.

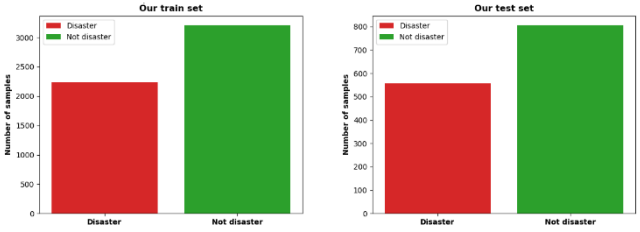
E. *Δοκιμή με χρήση των λέξεων κλειδιών και τοποθεσιών*

Σε αυτό το βήμα γνωρίζοντας τον καλύτερο συνδυασμό δοκιμάσαμε να προσθέσουμε στην αρχή των προεπεξεργασμένων tweets του συνδυασμού τις λέξεις κλειδιά και τις τοποθεσίες. Έτσι δοκιμάσαμε προσθέτοντας μόνο τη λέξη κλειδί ή μόνο την τοποθεσία ή και τα δύο. Αν κάποιο από αυτά έβγαζε καλύτερο αποτέλεσμα στο test set, με τη διαδικασία της σημείωσης, από το αποτέλεσμα του βήματος D θα το επιλέγαμε ως τον τελικό μας συνδυασμό. Σε Tweets χωρίς λέξη κλειδί ή τοποθεσία δεν προσθέταμε κάτι.

VI. Αποτελεσματα Πειραματων

A. *Αποτελέσματα βήματος A*

Παρατηρούμε ότι ο διαχωρισμός των δεδομένων μετά την αφαίρεση των διπλοτύπων μας έδωσε 5447 tweets στο train set και 1362 tweets στο test set. Όπως βλέπουμε στην εικόνα 7 στα δεδομένα έγινε ίση κατανομή ανάμεσα στα disaster και στα not disaster tweets.



Εικόνα 7. Κατανομή δεδομένων μετά το διαχωρισμό σε train και test sets.

B. *Αποτέλεσμα βήματος B*

Με βάση τα αποτελέσματα φάνηκε ότι από τις 8 μεθόδους προεπεξεργασίας που εξετάσαμε οι καλύτερες ήταν οι 'lemmatization', 'no_punc_no_abb ', 'ekphrasis' και 'ekphrasis_rm'. Στις μεθόδους προεπεξεργασίας 'no_punc_no_abb ', 'ekphrasis' και 'ekphrasis_rm' ο καλύτερος διανυσματοποιητής ήταν ο DistilBERT ενώ για τη μέθοδο 'lemmatization' ο καλύτερος διανυσματοποιητής ήταν TF-IDF με μέγιστο αριθμό χαρακτηριστικών 5000. Η μέθοδο προεπεξεργασίας DistilBERT λειτουργούσε πολύ καλύτερα συγκριτικά με τις υπόλοιπες μεθόδους προεπεξεργασίας και είχε καλύτερα αποτελέσματα όταν διατηρούσαμε τα σημεία στίξης και δεν κάναμε lemmatization ή stemming και ούτε αφαίρεση stop words. Στη μέθοδο προεπεξεργασίας DistilBERT ο αλγόριθμος μηχανικής μάθησης Logistic Regression παρουσίαζε πολύ καλά αποτελέσματα καθώς και ο αλγόριθμος SVC παρουσίαζε εξίσου καλά αποτελέσματα γι' αυτό αποφασίσαμε να διερευνήσουμε περεταίρω και τους 2 αλγόριθμους στο επόμενο βήμα. Οι αλγόριθμοι K-NN, Decision Tree και Random Forest είχαν χαμηλότερο score σε όλες τις περιπτώσεις και αποφασίσαμε να μην τους διερευνήσουμε περεταίρω. Στον πίνακα 1 μπορείτε να δείτε τα αποτελέσματα για αυτές τις μεθόδους προεπεξεργασίας. Περισσότερες πληροφορίες για τις μεθόδους προεπεξεργασίας υπάρχουν στο παράρτημα B.

Preprocessing	**10-folds Cross validation**		**Test set**
	Mean F1-weighted	***F1-weighted Std***	***F1-weighted***
lemmatization	0.778	0.017	0.803
no_punc_no_abb	0.801	0.016	0.810
ekphrasis	0.807	0.013	0.826
ekphrasis_rm	0.806	0.006	0.819

Πίνακας 1. Αποτελέσματα μετρήσεων του βήματος B για τις 4 πιο υποσχόμενες μεθόδους προεπεξεργασίας. Οι μέθοδοι διανυσματοποίησης και μηχανικής μάθησης είναι για όλες Logistic Regression, η πρώτη TF-IDF και οι τρεις τελευταίες DistilBERT .

C. *Αποτέλεσμα βήματος C*

Για τον αλγόριθμο Logistic Regression όπως και για τον αλγόριθμο SVC που είχε εξίσου καλά αποτελέσματα στη μέθοδο DistilBERT εξετάσαμε διάφορες παραμέτρους με την ελπίδα βελτίωσης των αποτελεσμάτων. Τα μοντέλα που χρησιμοποιήθηκαν είναι τα 4 που αναφέρονται στον πίνακα 1. Αυτό έγινε αυτόματα με την αναζήτηση πλέγματος (grid search) που προσφέρει η sklearn. Η αναζήτηση πλέγματος μας επέστρεψε τις παραμέτρους με το καλύτερο αποτέλεσμα. Για τον αλγόριθμο Logistic Regression παρουσιάζονται στον πίνακα 2 και για τον αλγόριθμο SVC στον πίνακα 3

Preprocessing	Logistic Regression	
	Best Hyperparameters	
	C	*max_iter*
lemmatization	1.6237	50
no_punc_no_abb	0.0885	50
ekphrasis	0.2335	200
ekphrasis_rm	0.6158	100

Πίνακας 2. Παράμετροι που επιστράφηκαν από την αναζήτηση πλέγματος (grid search). Ο αλγόριθμος μηχανικής μάθησης ήταν ο Logistic regression και η μέθοδος διανυσματοποίησης είναι για την πρώτη TF-IDF και τις τρεις τελευταίες DistilBERT.

Preprocessing	SVC		
	Best Hyperparameters		
	C	*gamma*	*kernel*
lemmatization	1	0.01	linear
no_punc_no_abb	10	0.01	rbf
ekphrasis	10	0.01	rbf
ekphrasis_rm	10	0.01	rbf

Πίνακας 3. Παράμετροι που επιστράφηκαν από την αναζήτηση πλέγματος (grid search). Ο αλγόριθμος μηχανικής μάθησης ήταν ο SVC και η μέθοδος διανυσματοποίησης είναι για την πρώτη TF-IDF και τις τρεις τελευταίες DistilBERT.

Ακόλουθος εξετάζοντας τις πιο πάνω παραμέτρους κάνωντας 10-folds cross validation το καλύτερο score στο test set είχε η μέθοδος προεπεξεργασίας ekphrasis με το ζεύγος διανυσματοποιητή DistilBERT και αλγόριθμο μηχανικής μάθησης SVC. Πιο συγκεκριμένα το cross validation είχε mean f1-weighted score ίσο με 0.811, τυπική απόκλιση ίση με 0.01399 και το test είχε f1-weighted score 0.837.

D. Αποτέλεσμα βήματος D

Γνωρίζοντας τις καλύτερες παραμέτρους για τη μέθοδο προεπεξεργασίας ekphrasis με ζεύγος διανυσματοποιητή DistilBERT και αλγόριθμο μηχανικής μάθησης SVC επαναλάβαμε τη διαδικασία της αναζήτησης πλέγματος διερευνώντας πιο πολλές τιμές του C κοντά στο 10 και πιο πολλές τιμές του gamma κοντά στο 0.01. Διατηρήσαμε τον kernel rbf σταθερό εφόσον αυτός επιλέχθηκε σαν καλύτερος σε όλες τις περιπτώσεις που χρησιμοποιήθηκε DistilBERT. Σε αυτή τη φάση η αναζήτηση πλέγματος μας επέστρεψε σαν βέλτιστες παραμέτρους τις τιμές C=11 και gamma=0.01.

E. Αποτέλεσμα βήματος E

Γνωρίζοντας πλέον τον καλύτερο συνδυασμό αποφασίσαμε να δοκιμάσουμε να προσθέσουμε στην αρχή του κάθε tweet την λέξη κλειδί ή την τοποθεσία ή και τα δυο, αν υπήρχαν. Εάν και περιμέναμε τουλάχιστον οι λέξεις κλειδιά να δώσουν κάποια πληροφορία τελικά καμία προσπάθεια δεν βελτίωσε το f1-weighted score στο test set. Στον πίνακα 4 παρουσιάζονται τα πειραματικά αποτελέσματα. Μια άλλη παρατήρηση είναι πως οι τοποθεσίας χειροτερεύουν το σκορ και επομένως σωστή ήταν η υπόθεση μας πως έπρεπε να αγνοηθούν.

Κάτι στο οποίο δεν αναφερθήκαμε είναι πως την στήλη id στα δεδομένα την αγνοήσαμε από την αρχή αφού δε μας δίνει καμία πληροφορία και ούτε μπορούμε να τη χρησιμοποιήσουμε προς όφελος μας.

DistilBERT with SVC(C = 11, gamma = 0.01, kernel = rbf)			
Preprocessing	10-folds Cross validation		Test set
	Mean F1-weighted	*F1-weighted Std*	*F1-weighted*
ekphrasis	0.814	0.011	0.831
keyword_ekphrasis	0.814	0.016	0.820
location_ekphrasis	0.801	0.016	0.823
keyword_location_ekphrasis	0.807	0.012	0.821

Πίνακας 4. Αποτελέσματος πειράματος για το βήμα E.

VII. Τελικά Αποτελεσματα

Στο τελικό μας αποτέλεσμα καταλήξαμε με τον ακόλουθο συνδυασμό. Για μέθοδο διανυσματοποίησης επιλέχθηκε η μέθοδος DistilBERT. Για αλγόριθμο μηχανικής μάθησης επιλέχθηκε ο SVC με C ίσο με 11 , gamma ίσο με 0.01 , kernel ίσο με rbf, όλα τα υπόλοιπα ήταν τα default. Για τη μέθοδο προεπεξεργασίας επιλέχθηκε η μέθοδος ekphrasis. Μετά από την εφαρμογή 10-folds cross validation ο μέσος όρος f1-weighted score ήταν 81.05% (+/- 1.66%). Το f1-weighted score στο test set ήταν 83.14%. Το αποτέλεσμα ήταν αρκετά αξιόπιστο αφού η τυπική απόκλιση στο cross validation δεν είναι μεγάλη και επιπλέον το σκορ στο test set είναι αρκετά κοντά στο αποτέλεσμα που περιμέναμε.

Τέλος δημιουργήσαμε προβλέψεις στο test set που διατηρεί το Kaggle και κάναμε αποστολή των προβλέψεων του καλύτερου μας συνδυασμού. Το αποτέλεσμα που πήραμε ήταν 82.208%.

VIII. Μελοντικη Εργασια

Στο μέλλον θα μπορούσαν να γίνουν εκτενέστεροι έλεγχοι των δυνατών παραμέτρων των διάφορων αλγόριθμων μηχανικής μάθησης, όπως και των διάφορων διανυσματοποιητών. Επίσης μεγάλες δυνατότητες για δημιουργία νευρωνικών δικτύων προσφέρονται μέσα από τις βιβλιοθήκες του tensorflow και keras[6]. Ένα είδος νευρωνικών δικτύων που ίσως να έδινε καλύτερα αποτελέσματα είναι τα recurrent neural networks (RNN). Αυτό αφήνεται για μελλοντική εργασία

Βιβλιογραφια

[1] Beautiful Soup Documentation — Beautiful Soup 4.9.0 Documentation. [online] Available at: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/> [Accessed 17 April 2020].

[2] GitHub. 2020. Cbaziotis/Ekphrasis. [online] Available at: <https://github.com/cbaziotis/ekphrasis> [Accessed 17 April 2020].

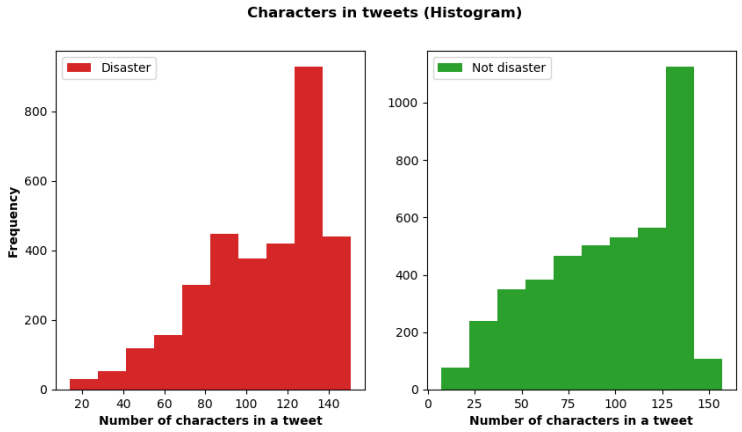
[3] GitHub. 2020. Google-Research/Bert. [online] Available at: <https://github.com/google-research/bert> [Accessed 17 April 2020].

[4] Distilbert — Transformers 2.8.0 Documentation. [online] Available at: <https://huggingface.co/transformers/model_doc/distilbert.html> [Accessed 17 April 2020].

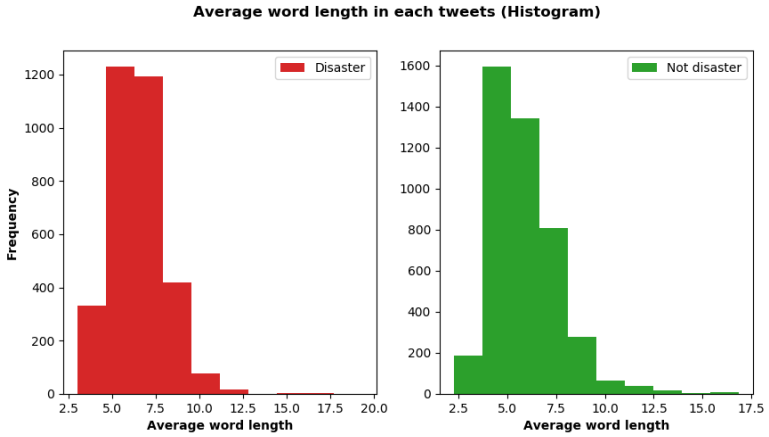
[5] Scikit-Learn: Machine Learning In Python — Scikit-Learn 0.22.2 Documentation. [online] Available at: <https://scikit-learn.org/stable/> [Accessed 17 April 2020].

[6] TensorFlow. 2020. Module: Tf.Keras | Tensorflow Core V2.1.0. [online] Available at: <https://www.tensorflow.org/api_docs/python/tf/keras> [Accessed 17 April 2020].

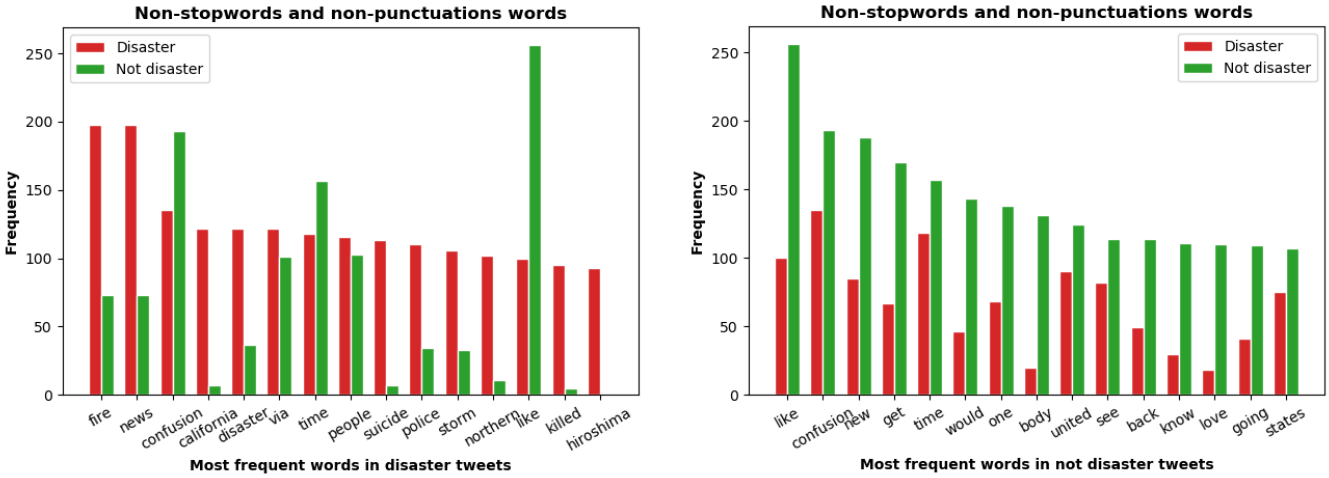
ΠΑΡΑΡΤΗΜΑ Α – ΕΠΙΠΡΟΣΘΕΤΕΣ ΓΡΑΦΙΚΕΣ ΠΑΡΑΣΤΑΣΕΙΣ



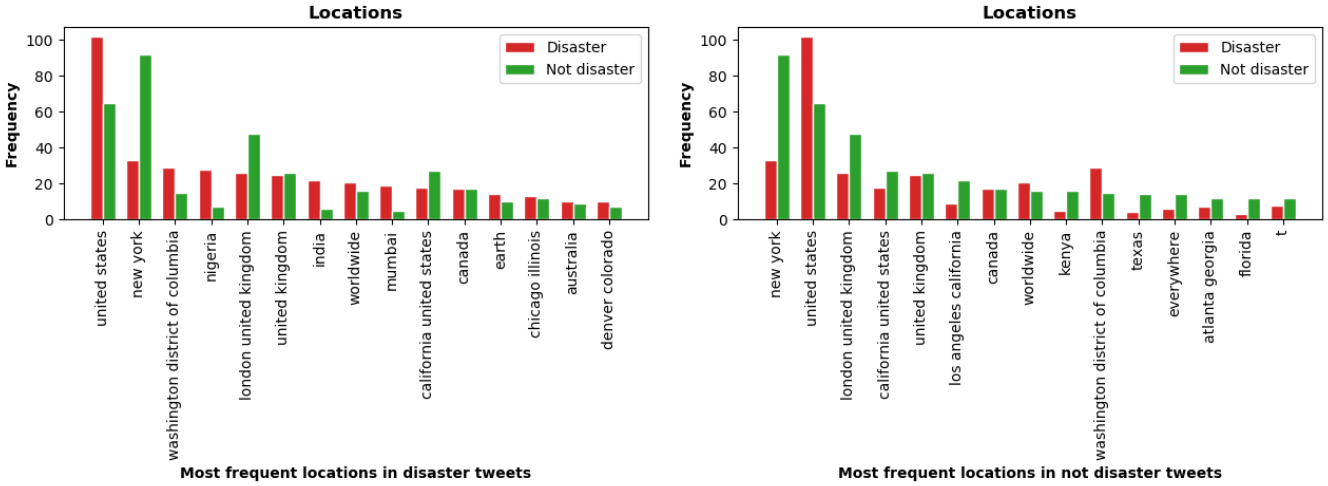
Εικόνα Α1. Πλήθος χαρακτήρων στα tweets



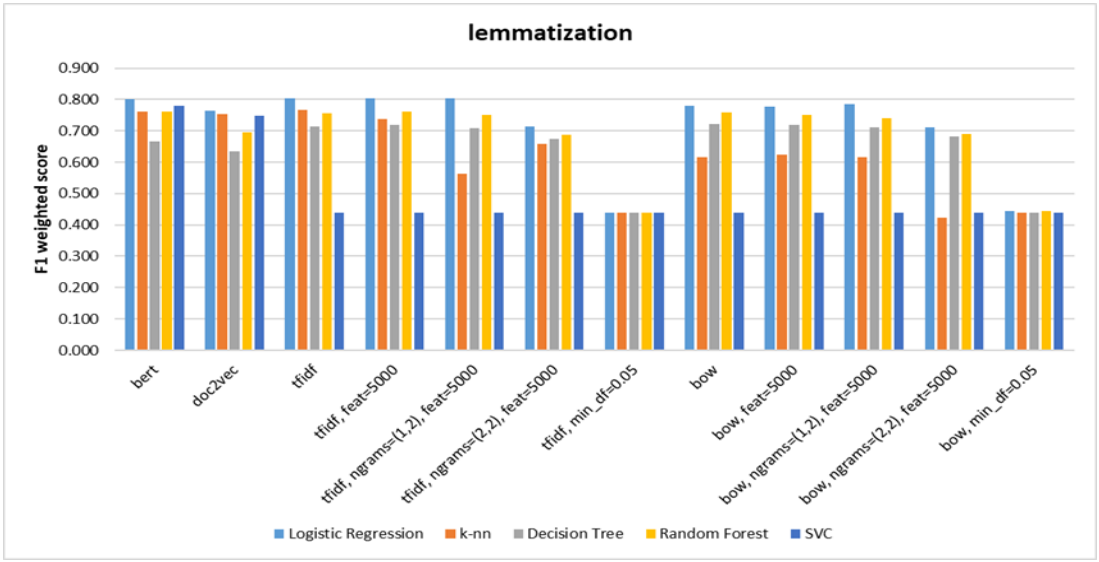
Εικόνα Α2. Μέσο μήκος λέξεων στα tweets



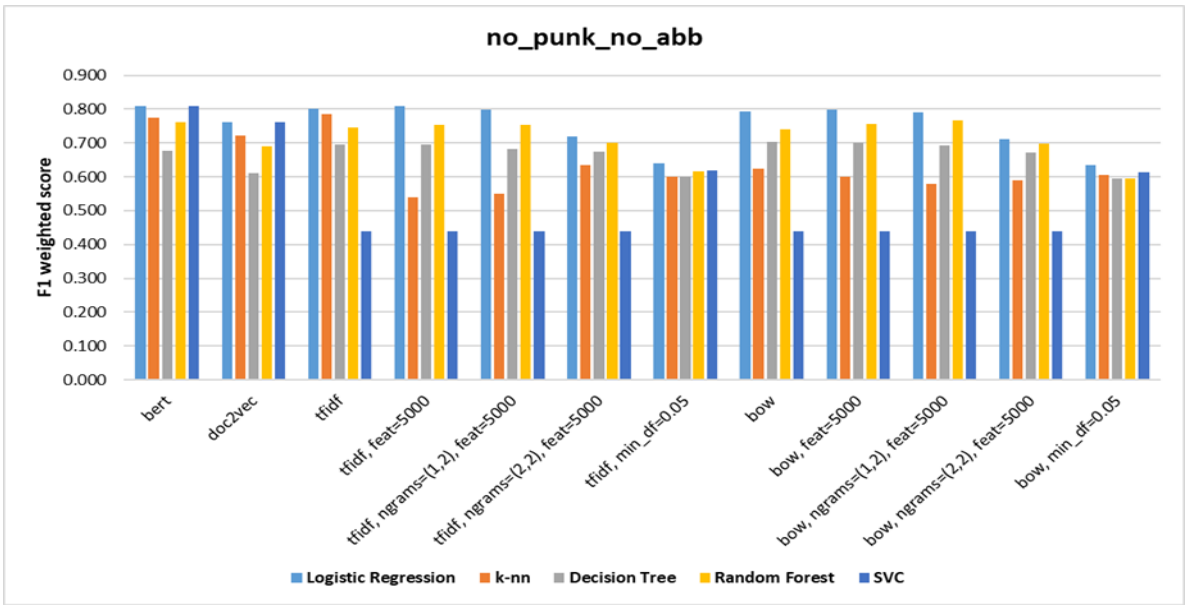
Εικόνα Α3. Πιο συχνές λέξεις στα tweets (non-stopwords και non-punctuations)



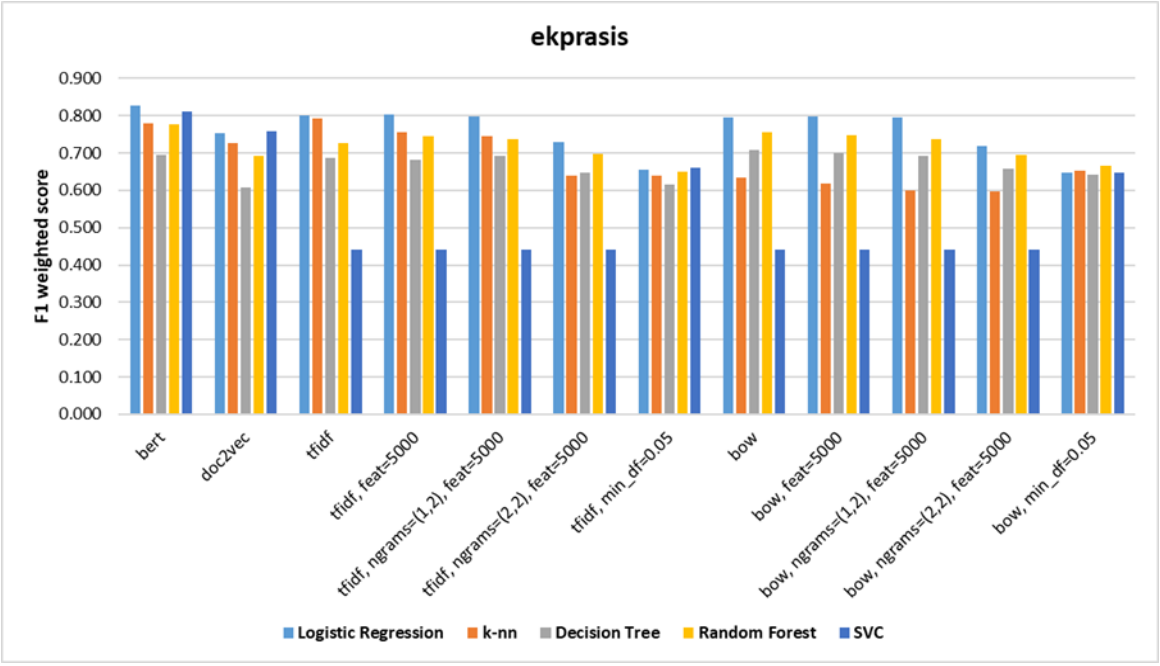
Εικόνα Α4. Πιο συχνές τοποθεσίες στα tweets



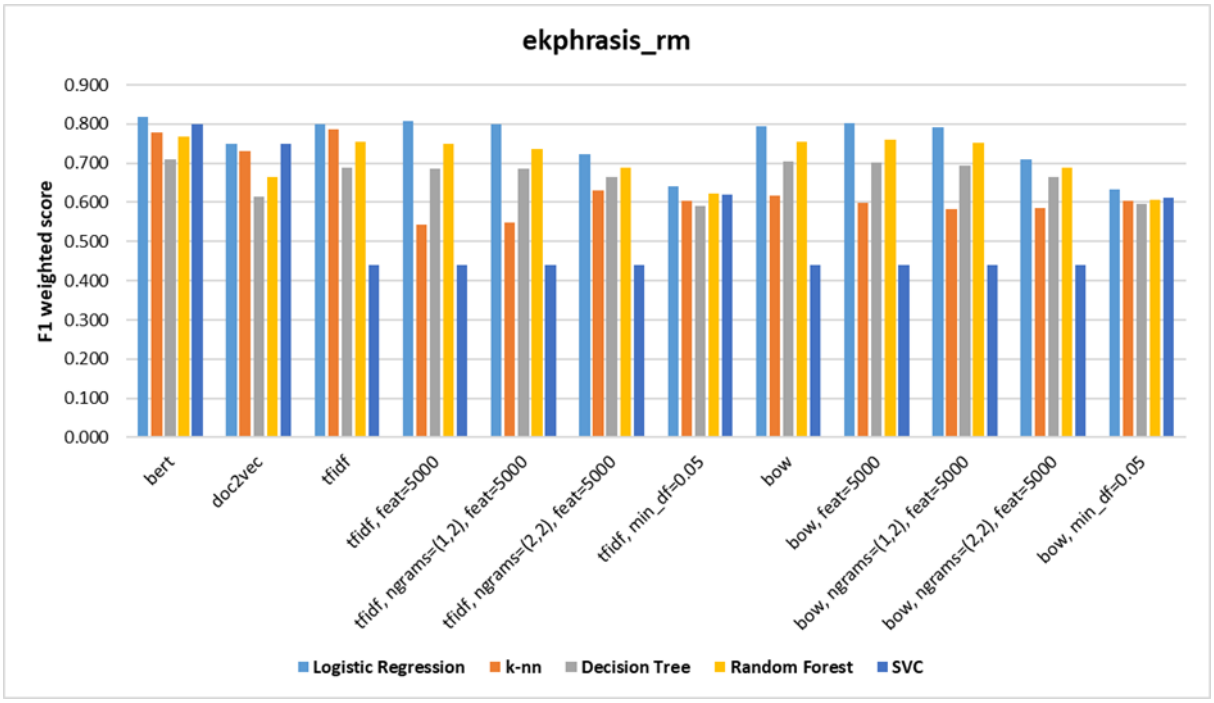
Εικόνα Α5. F1-weighted score για το test set για όλα τα ζεύγη διανυσματοποιητή και αλγόριθμων μηχανικής μάθησης για τη μέθοδο προεπεξεργασίας ‘lemmatization’



Εικόνα Α6. F1-weighted score για το test set για όλα τα ζεύγη διανυσματοποιητή και αλγόριθμων μηχανικής μάθησης για τη μέθοδο προεπεξεργασίας ‘no_punc_no_abb’



Εικόνα A7. F1-weighted score για το test set για όλα τα ζεύγη διανυσματοποιητή και αλγόριθμων μηχανικής μάθησης για τη μέθοδο προεπεξεργασίας ‘ekphrasis’



Εικόνα A8. F1-weighted score για το test set για όλα τα ζεύγη διανυσματοποιητή και αλγόριθμων μηχανικής μάθησης για τη μέθοδο προεπεξεργασίας ‘ekphrasis_rm’

ΠΑΡΑΡΤΗΜΑ Β – ΠΡΟΕΠΕΞΕΡΓΑΣΙΑ ΔΕΔΟΜΕΝΩΝ

Σε όλες τις στήλες που παρουσιάζονται στον πίνακα Β.1 εφαρμόστηκε η ακόλουθη προεπεξεργασία:

- διαγραφή URLs
- αντικατάσταση html χαρακτήρων με τους αντίστοιχους Unicode χαρακτήρες
- διαγραφή usernames
- αντικατάσταση emojis & emoticons με την εμρηνεία τουυς
- διαγραφή non ascii και non printable χαρακτήρων
- διαγραφή συνεχόμενων επαναλαμβανόμενων χαρακτήρων
- μετατροπή χαρακτήρων σε lowercase και διαχωρισμός λέξεων υπό μορφή camelCase
- επέκταση hashtags
- αντικατάσταση συντομογραφιών (contractions)
- διαγραφή αριθμών numbers

	Applied Preprocessing	**Comments**
keyword	1. αντικατάσταση χαρακτήρα %20 με κενό	-
location_processed	1. αντικατάσταση συντομογραφιών με χρήση λεξικού για ονόματα πολιτειών των ΗΠΑ και άλλων κύριων χωρών	-
Preprocessing method		
no_punc_no_abb	-	-
lemmatization	1. διαγραφή stopwords 2. lemmatization	-
ekphrasis	1. normalization and annotation (URL, email, percent, money, phone, user, time, date, number) 2. fix html 3. χρήση social tokenizer	χρήση βιβλιοθήκης ekphrasis
ekphrasis_rm	1. διαγραφεί συμβόλων '<' και '>' από τα tags που προκύπτουν από την διαδικασία του normalization/annotation	εφαρμογή σε δεδομένα ekphrasis
keyword_ekphrasis	1. πρόσθεση keyword στην αρχή των δεδομένων στήλης ekprhasis, εάν υπάρχει	-
location_ekphrasis	1. πρόσθεση keyword στην αρχή των δεδομένων στήλης ekprhasis, εάν υπάρχει	-
keyword_location_ekphrasis	1. πρόσθεση keyword και location_processed στην αρχή των δεδομένων στήλης ekprhasis, εάν υπάρχει	-

Πίνακας Β1. Επιπλέον εφαρμοσμένη προεπεξεργασία ανά στήλη (μέθοδο προεπεξεργασίας)