Εργασία για το μάθημα Τεχνικές Εξόρυξης Δεδομένων

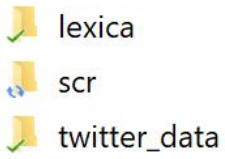
Εαρινό εξάμηνο , Ακ. Έτος 2018-19

Ανάλυση συναισθημάτων (Sentiment Analysis): η διαδικασία υπολογιστικής ταυτοποίησης και κατηγοριοποίησης των απόψεων που εκφράζονται σε ένα κομμάτι κειμένου, προκειμένου να καθοριστεί εάν η στάση του συγγραφέα έναντι ενός συγκεκριμένου θέματος, προϊόντος κλπ. είναι θετική, αρνητική ή ουδέτερη.

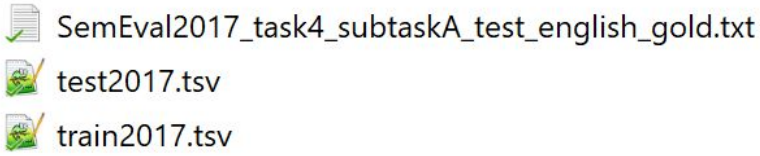
Τα δεδομένα στα οποία θα εργαστείτε προέρχονται από τον ετήσιο διαγωνισμό SemEval (International Workshop on Semantic Evaluation) και συγκεκριμένα το διαγωνισμό του έτους 2017 που αφορούσε ανάλυση συναισθήματος σε tweets.

Δεδομένα για την εργασία:

Στο eclass θα βρείτε ένα φάκελο για την εργασία σας. Ο φάκελος αυτός έχει την παρακάτω δομή:



Στο φάκελο **twitter_data** περιέχονται 3 αρχεία,



Το αρχείο **train2017.tsv** περιέχει τα δεδομένα που θα χρησιμοποιήσετε για εκπαίδευση των μοντέλων σας. Τα δεδομένα εκπαίδευσης περιέχουν 28061 tweets με την ένδειξη positive, negative ή neutral.

Το **test2017.tsv** περιέχει τα δεδομένα που θα χρησιμοποιήσετε για να δοκιμάσετε το μοντέλο σας και να κάνετε μία πρόβλεψη. Τα δεδομένα δοκιμής περιέχουν 12284 tweets με την ένδειξη UNKNOWN , καθώς για αυτό το σύνολο των tweets πρέπει το μοντέλο σας να αποφασίσει αν εφράζει θετικό, αρνητικό ή ουδέτερο συναίσθημα.

Τέλος το αρχείο *.gold.txt περιέχει τα σωστά labels για το αρχείο test2017 . Τα gold labels δεν επιτρέπεται να χρησιμοποιηθούν σε καμία περίπτωση στην εκπαίδευση των μοντέλων σας. Μπορείτε μόνο να τα χρησιμοποιήσετε για την επαλήθευση των αποτελεσμάτων σας και συγκεκριμένα μπορείτε να υπολογίσετε πόσο καλά τα πηγαίνει ο ταξινομητής σας (για παράδειγμα μπορείτε να χρησιμοποιήσετε το F1 score https://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html).

Στο φάκελο **lexica** σας δίνουμε διάφορα συναισθηματικά λεξικά που μπορείτε να χρησιμοποιήσετε για την εργασία σας. Τα λεξικά αυτά όπως θα δείτε, περιλαμβάνουν για κάθε λέξη μία συνεχή τιμή στο διάστημα [-1,1], η οποία αντιπροσωπεύει αυτό που στην συναισθηματική ανάλυση ονομάζουμε “valence” μιας λέξης (*means the intrinsic attractiveness/"good"-ness (positive valence) or averseness/"bad"-ness (negative valence) of an event, object, or situation*). Τα λεξικά έχουν προκύψει είτε με την μέθοδο των αξιολογητών είτε με μεθόδους μηχανικής μάθησης. Στα λεξικά θα βρείτε αρκετές από τις λέξεις που περιέχονται στα tweets. Μπορείτε να χρησιμοποιήσετε όποια από αυτά τα λεξικά θέλετε, ή ακόμα και να βρείτε άλλα και να τα συνδυάσετε.

Ζητούμενα:

Η εργασία θα γίνει με την γλώσσα προγραμματισμού Python. Στη σχετική ενότητα στο eclass θα βρείτε όλο το υλικό που αφορά την python.

Προεπεξεργασία και καθάρισμα των δεδομένων

a. Καθάρισμα των δεδομένων, (αφαιρούμε τα σύμβολα, όπως hashtags, emoticons,emojis, τα links και τα stopwords από το training set)
b. tokenization
c. stemming

Για τα b, c μπορείτε να χρησιμοποιήσετε το NLTK Python toolkit ή όποια άλλη βιβλιοθήκη θέλετε.

Ανάλυση των δεδομένων.

Καλείστε να γράψετε μερικές εντολές σε python που θα σας βοηθήσουν να “μελετήσετε” τα δεδομένα που σας δίνονται και να εξάγετε μερικά συμπεράσματα. Μερικά από τα ερωτήματα που μπορείτε να απαντήσετε είναι τα παρακάτω:

Ποιες είναι οι συνηθέστερες λέξεις σε ολόκληρο το σύνολο δεδομένων;
Ποιες είναι οι συνηθέστερες λέξεις στο σύνολο δεδομένων για αρνητικά, θετικά και ουδέτερα tweets, αντίστοιχα;

Μπορείτε να παρουσιάσετε τα παραπάνω αποτελέσματα με ένα Word cloud. Μπορείτε να σκεφτείτε και κάποιες άλλες παρατηρήσεις που προκύπτουν από τα δεδομένα ; Αν είναι εφικτό παρουσιάστε σχετικά γραφήματα.

Vectorization - εξαγωγή χαρακτηριστικών

Ακολουθήστε τις οδηγίες που παρουσιάσαμε στο φροντιστήριο και ετοιμάστε τα χαρακτηριστικά για κάθε tweet χρησιμοποιώντας:

1. Bag-of-words
2. Tf-idf
3. word embeddings

Χρησιμοποιήστε τη βιβλιοθήκη pickle της Python για να αποθηκεύσετε τα χαρακτηριστικά σε αρχεία *.pkl . Με αυτό τον τρόπο δεν χρειάζεται να υπολογίζονται από την αρχή τα χαρακτηριστικά κάθε φορά που τρέχετε το πρόγραμμά σας, αλλά μπορείτε μόνο να τα φορτωνεται στην μνήμη χρησιμοποιώντας την αντίστοιχη μέθοδο *load*.

Προσθήκη χαρακτηριστικών στο διάνυσμα λέξης

Με χρήση των embeddings παίρνουμε για κάθε tweet ένα διάνυσμα με 200-300 τιμές (features). Μπορούμε να “προσθέσουμε” επιπλέον χαρακτηριστικά επεκτείνοντας το παραπάνω πίνακα. Για το σκοπό αυτό μπορείτε να χρησιμοποιήσετε τα λεξικά που αντιστοιχούν τιμές (συνεχείς) συναισθήματος, σε λέξεις. Για κάθε tweet αναζητείστε τις αντίστοιχες λέξεις του στα λεξικά και υπολογίστε μία μέση τιμή “συναισθήματος” σε όλο

το tweet, ανά λεξικό. Μετά από αυτό το βήμα θα έχετε για παράδειγμα ένα διάνυσμα για κάθε tweet μεγέθους [300 + N], όπου N το πλήθος των λεξικών που χρησιμοποιήσατε. **Bonus**: Μπορείτε να σκεφτείτε και άλλα χαρακτηριστικά που θα μπορούσαν να προστεθούν στο διάνυσμα της λέξης αυξάνοντας το πλήθος των χαρακτηριστικών ; (τέτοια παραδείγματα είναι: μήκος του tweet,μέγιστη και ελάχιστη τιμή του valence των λέξεων σε κάθε tweet, να χωρίσουμε το tweet σε δύο μέρη και να υπολογίσουμε μέση τιμή valence στο πρώτο και στο 2ο μισό κ.α.)

Δοκιμάζουμε ταξινομητές (SVM, KNN, Round Robin Classification)

a. SVM
b. KNN

Δοκιμάστε τους ταξινομητές σας με τα χαρακτηριστικά BOW, TFID, word embeddings και με το διάνυσμα στο οποίο έχετε προσθέσει τα features από τα λεξικά.

c. Round Robin Classification - **bonus** : Το αρχικό σας πρόβλημα είναι ένα πρόβλημα ταξινόμησης σε 3 κατηγορίες. Μπορείτε όμως να δοκιμάσετε την τεχνική του pairwise classification, όπως περιγράφεται στην παρακάτω δημοσίευση.
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.14.5074&rep=rep1&type=pdf

Η round robin ταξινόμηση είναι μια τεχνική που χωρίζει το πρόβλημα σε επιμέρους δυαδικά προβλήματα. Στην συνέχεια χρησιμοποιεί ένα ταξινομητή για κάθε ζεύγος κλάσεων. Για την μέθοδο αυτή θα χρησιμοποιήσετε Nearest neighbor classifier.

Ο αλγόριθμος είναι ο εξής:

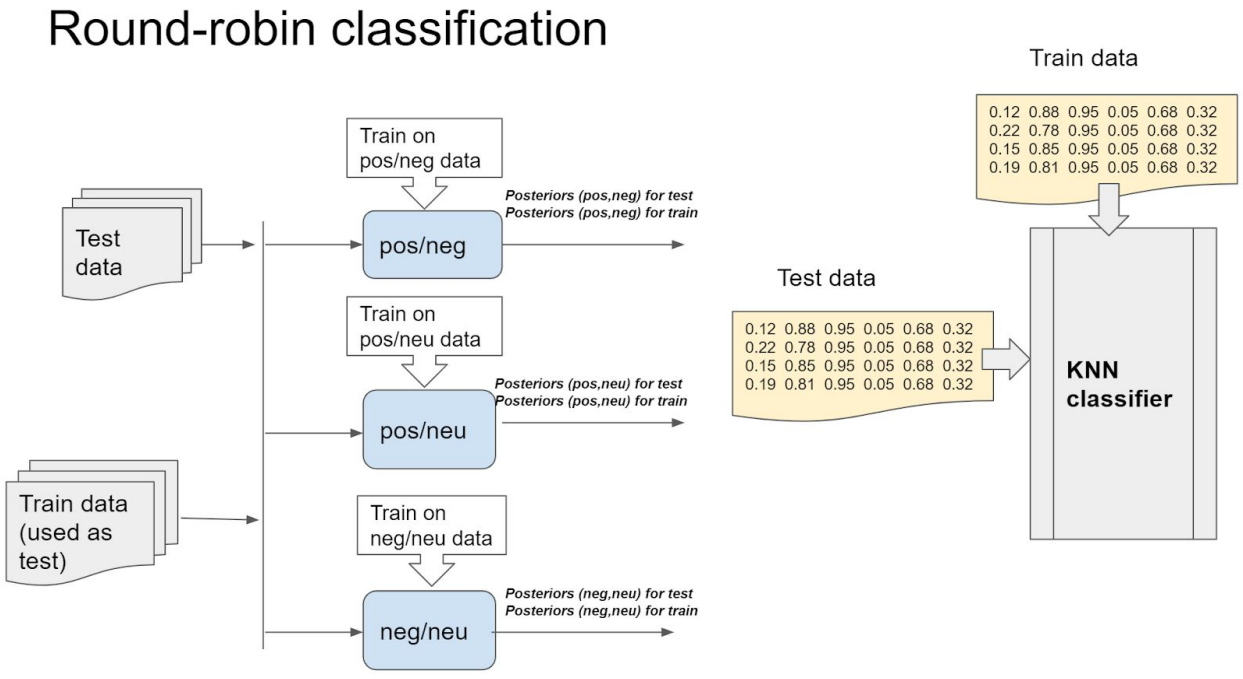
I. Μετατρέπω το πρόβλημα κατηγορίας **c** σε **c(c-1) / 2** προβλήματα δύο κατηγοριών, ένα για κάθε σύνολο κλάσεων {i, j}, i = 1 ... c-1, j = i + 1 ... c.
II. Ο δυαδικός ταξινομητής εκπαιδεύεται με παραδείγματα κλάσεων i και j.
III. Τα παραδείγματα των υπόλοιπων κλάσεων που δεν ανήκουν στις i, j αγνοούνται.
IV. Τα αποτελέσματα των επιμέρους ταξινομητών συνδυάζονται σε ένα ξεχωριστό ταξινομητή.

Στο δικό μας πρόβλημα, έχουμε 3 κατηγορίες, *positive,negative,neutral*. Επομένως θα έχουμε 3*(3-1)/2 = 6 διαφορετικά προβλήματα, τα οποία είναι για παράδειγμα, positive-negative, negative-neutral, positive-neutral κτλ. Συνεπώς έχουμε 6 διαφορετικούς ταξινομητές. Η έξοδος κάθε ταξινομητή μπορεί να είναι είτε το class label, δηλαδή να επιστρέψει positive, negative,neutral, είτε το posterior , δηλαδή να

αντιστοιχίσει μία τιμή πιθανότητας σε κάθε κατηγορία (δείτε στο sklearn την μέθοδο **predict_proba**).

Ας πάρουμε ένα απλό παράδειγμα. Ο πρώτος ταξινομητής (positive-negative) θα εκπαιδευτεί (μέθοδος fit) μόνο στα tweets που είναι positive ή negative. Μπορείτε να χρησιμοποιήσετε σε αυτό το ερώτημα όποιο από τα διανύσματα χαρακτηριστικών έχετε αποθηκεύσει στα προηγούμενα ερωτήματα. Στη συνέχεια θα δοκιμαστεί (μέθοδος predict) σε όλα τα tweets του train set. Η έξοδος του ταξινομητή που παράγει posterior probability έχει 2 τιμές, για κάθε tweet της εισόδου του. Θα εφαρμόσουμε την μέθοδο predict και σε όλα τα test tweets σε αυτό τον ταξινομητή. Με τη ίδια διαδικασία, κάθε διαδικός ταξινομητής δοκιμάζεται όχι μόνο στα test αλλά και στα train δεδομενα ώστε να πάρουμε τα αντίστοιχα διανύσματα χαρακτηριστικών.

Σχηματικά ο αλγόριθμος φαίνεται παρακάτω (στο τελευταίο βήμα ο ταξινομητής KNN δέχεται έναν πίνακα με 6 χαρακτηριστικά για κάθε tweet, τα οποία προέρχονται από το αποτέλεσμα (posterior) των επιμέρους ταξινομητών):



Παρουσίαση αποτελεσμάτων:

Σχεδιάστε έναν πίνακα στο οποίο να φαίνονται τα αποτελέσματά σας για τους διαφορετικούς ταξινομητές που χρησιμοποιήσατε με βάση τα διαφορετικά διανύσματα

χαρακτηριστικών. Σχολιάστε τα αποτελέσματά σας (πότε παρατηρείται βελτίωση, ποιά πιστεύετε ότι είναι τα καλύτερα χαρακτηριστικά κα) .

Παραδοτέο:

Η εργασία μπορεί να εκπονηθεί ατομικά ή σε ομάδες **2 ατόμων**.

Ο φάκελος **scr** της εργασίας είναι ο φάκελος στον οποίο θα γράψετε τον κώδικά σας και είναι και αυτός που θα παραδώσετε (δηλαδή δεν θα παραδώσετε εκ νέου τα λεξικά και τα δεδομένα εκπαίδευσης/δοκιμής). Θα ανεβάσετε στο eclass ένα φάκελο της μορφής scr_sdixxxx. (όπου sdi το ΑΜ ενός εκ των ατόμων της ομάδας).

Ο κώδικάς σας πρέπει να περιέχει ΥΠΟΧΡΕΩΤΙΚΑ ένα **Ipython notebook** με το οποίο θα μπορεί κάποιος να τρέξει την εργασία σας βήμα-βήμα. Μπορείτε να έχετε και *.py αρχεία με τις συναρτήσεις σας αλλά η εργασία πρέπει να τρέχει από ένα notebook. Στο notebook μπορείτε σε όποια σημεία κρίνετε απαραίτητο να εισάγετε **visualizations** με τον τρόπο που θα εξηγήσουμε στα φροντιστήρια (ενδεικτικά αναφέρουμε το word cloud, word embedding visualization και φυσικά μπορείτε να παρουσιάσετε και με ωραίο τρόπο τα αποτελέσματά σας). **Το notebook αποτελεί και την ολοκληρωμένη αναφορά** για την εργασία σας (δεν θα παραδώσετε τίποτα σε doc, pdf) , σχεδιάστε το με προσοχή, να θυμάστε να γράψετε μία περιγραφή σε κάθε βήμα για το τι κάνει ο κώδικάς σας σε κάθε κελί.