Τεχνικές Εξόρυξης Δεδομένων
2η Άσκηση

Ατομική ή Ομαδική Εργασία (2 Ατόμων) (σημείωση: *Αν πρόκειται για ομαδική εργασία να χρησιμοποιηθεί η ίδια ομάδα με την 1η Άσκηση)*

ΠΑΡΑΔΟΣΗ: Κυριακή, 9 Ιουνίου 2024, ώρα 23:55

Στην εργασία αυτή θα χρησιμοποιήσετε τα ίδια δεδομένα με αυτά της Άσκησης 1.

Ερώτημα 1: Study over time (40%)

Στο ερώτημα αυτό θα γίνει χρήση του Hugging Face για την επισημείωση προτάσεων από το σύνολο των δεδομένων που σας έχουν δοθεί. Μέσω του Hugging Face θα έχετε εύκολη πρόσβαση σε προ-κατασκευασμένα μοντέλα για την επισημείωση προτάσεων, όπως BERT, GPT, RoBERTa και πολλά άλλα. Τα μοντέλα αυτά έχουν εκπαιδευτεί σε μεγάλα σύνολα δεδομένων και έχουν εξελιχθεί στην επισημείωση προτάσεων παρέχοντας υψηλή απόδοση. Για τους σκοπούς της εργασίας θα χρησιμοποιήσουμε την στήλη comments από το αρχείο reviews. Αν δεν έχει ήδη γίνει θα χρειαστεί σε αυτή τη στήλη προεπεξεργασία. Αφαιρούμε τα σημεία στίξης, μετατρέπουμε όλους τους χαρακτήρες σε μικρούς, αφαιρούμε σύμβολα, όπως hashtags, emoticons,emojis, links και τα stopwords από το σύνολο των δεδομενων). Μπορείτε να χρησιμοποιήσετε οποιοδηποτε HuggingFace μοντέλο αφορά sentiment analysis. Ενδεικτικά αναφέρουμε *sentiment-roberta-large-english, distilbert-base-multilingual-cased-sentiments-student, BERT-Sentiment-Classifier* κα.

- Χρησιμοποιώντας το HuggingFace επισημείωστε (annotation process) ως προς το συναίσθημα (θετικό/αρνητικό/ουδέτερο) όσα περισσότερα comments μπορείτε για το 2019- *hint: μπορείτε να χωρίσετε το dataset σε μικρότερα κομμάτια (chunks)* και να γίνεται η επισημείωση τμηματικά . Στο τελικό αποτέλεσμα θα χρειαστεί να έχετε και ένα αναγνωριστικό id. Δηλαδή θα προκύψει ένα csv αρχείο (ή ένα dataframe) που θα έχει 3 στήλες (id, review, sentiment)
- Κάντε το ίδιο για τις κριτικές του 2023.
- Συγκρίνετε το συνολικό συναίσθημα με την πάροδο του χρόνου (πχ ένα ιστόγραμμα για κάθε χρόνο με την κατανομή των positive/negative/neutral).
- Συγκρίνετε το συναίσθημα ανά γειτονιά με την πάροδο του χρόνου (***BONUS - 5%***).

Ερώτημα 2: Sentiment Analysis (45%)

Στο ερώτημα αυτό θα χρειαστεί να δημιουργήσετε δύο νέα dataset από τα δεδομένα που επισημειώσατε στο προηγούμενο ερώτημα. Πιο συγκεκριμένα θα χρειαστείτε

- Ένα αρχείο train.tsv (θα είναι το 80% των συνολικών data points) που θα περιέχει τα δεδομένα που θα χρησιμοποιήσετε για εκπαίδευση των μοντέλων σας. Τα δεδομένα εκπαίδευσης περιέχουν την ένδειξη positive, negative ή neutral.
- Ενα αρχείο test.tsv (το 20% των data points) το οποίο θα περιέχει τα δεδομένα που θα χρησιμοποιήσετε για να δοκιμάσετε το μοντέλο σας και να κάνετε μία πρόβλεψη. Πρέπει το μοντέλο σας να αποφασίσει για κάθε ένα από τα reviews που υπάρχουν στο σύνολο των test αν εφράζει θετικό, αρνητικό ή ουδέτερο συναίσθημα.

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

- Tf-idf
- 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.

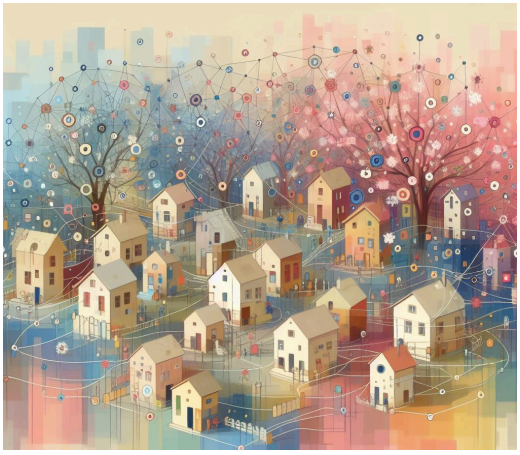
Δοκιμάζουμε τους παρακάτω ταξινομητές (η δοκιμή/testing θα γίνει στα test δεδομένα που έχουμε κρατήσει- εννοείται ότι τα test δεν θα χρησιμοποιηθούν για το training!)

- SVM
- Random Forests
- KNN

Δοκιμάστε τους ταξινομητές σας με τα χαρακτηριστικά TFID, και word embeddings. Επίσης θα πρέπει να αξιολογήσετε και να καταγράψετε την απόδοση κάθε μεθόδου χρησιμοποιώντας 10-fold Cross Validation χρησιμοποιώντας τις παρακάτω μετρικές:

- Precision / Recall / F-Measure
- Accuracy

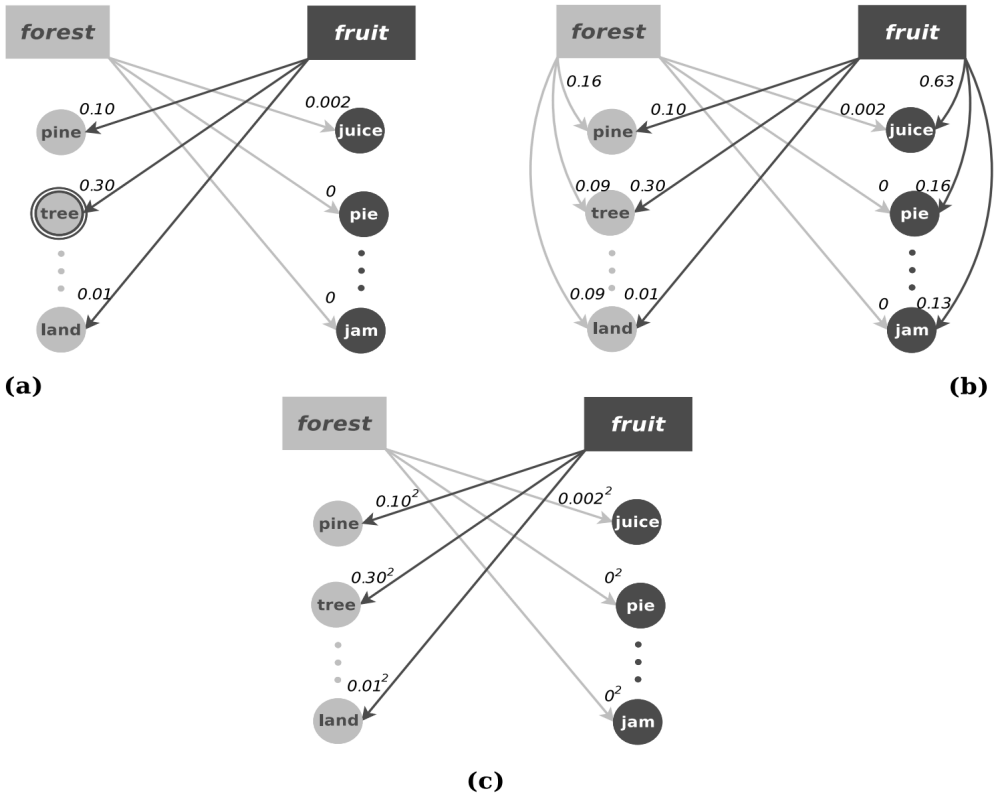
Ερώτημα 3: Similarity (jaccard, cosine, etc) και σημασιολογικές γειτονιές. (15%)



You Shall Know a Word by the Company It Keeps

Για τη στήλη comments και με χρήση word embeddings φτιάξτε μία συνάρτηση η οποία θα υπολογίζει το similarity μεταξύ vectors δύο λέξεων (πχ cosine ή όποια άλλη μέθοδο θέλετε, ενδεικτικά μπορείτε να δείτε μεθόδους σε αυτό τον σύνδεσμο) Μπορείτε να χρησιμοποιήσετε οποιαδήποτε μέθοδο φτιάχνει word embeddings (word2vec, fastText, glove κτλ) η να χρησιμοποιήσετε έτοιμα pre-trained embeddings.

Στη συνέχεια για δύο λέξεις (τυχαίες που θα δίνει ένας χρήστης) και για μία παράμετρο N να υπολογίζεται (α) η σημασιολογική γειτονιά των δύο λέξεων και (β) το similarity των λέξεων το οποίο θα προκύπτει από τη γειτονιά με βάση τα τρία παρακάτω σχήματα , δηλαδή 3 διαφορετικά similarities.



Pictorial view of neighborhood-based metrics. Two reference nouns, “forest” and “fruit”,are depicted along with their neighborhoods: {pine, tree, . . . , land} and {juice, pie, . .

. , jam}, respectively. Arcs represent the similarities between reference nouns and neighbors. The similarity between "forest" and "fruit" is computed according to (a) maximum similarity of neighborhoods, (b) correlation of neighborhood similarities, and (c) sum of squared neighborhood similarities.
https://slp-ntua.github.io/potam/preprints/journal/2013_NLE_senetwork_DSM_draft.pdf

$$M_n(w_i, w_j) = \max\{\alpha_{ij}, \alpha_{ji}\},$$

$$\alpha_{ij} = \max_{x \in N_j} S(w_i, x), \quad \alpha_{ji} = \max_{y \in N_i} S(w_j, y).$$

(a) maximum similarity of neighborhoods

$$R_n(w_i, w_j) = \max\{\beta_{ij}, \beta_{ji}\},$$

$$\beta_{ij} = \rho(C_i^{N_i}, C_j^{N_i}), \quad \beta_{ji} = \rho(C_i^{N_j}, C_j^{N_j})$$

$$C_i^{N_i} = (S(w_i, x_1), S(w_i, x_2), \ldots, S(w_i, x_n)), \quad \text{where } N_i = \{x_1, x_2, \ldots, x_n\}$$

(b) correlation of neighborhood similarities

Στο παράδειγμα του σχήματος, τα similarities της λέξης forest (C1) με την γειτονιά (N1) είναι

$$C_1^{N_1} = (0.16, 0.09, \ldots, 0.09)$$

Στο παράδειγμα του σχήματος, τα similarities της λέξης fruit(C2) με την γειτονιά (N1) είναι

$$C_2^{N_1} = (0.10, 0.30, \ldots, 0.01)$$

Ενώ αντίστοιχα υπολογίζονται τα $C_2^{N_1}$ και τα $C_1^{N_2}$. Το *ρ* είναι ο συντελεστής συσχέτισης pearson.
https://stackabuse.com/calculating-pearson-correlation-coefficient-in-python-with-numpy/

$$E_n^{\theta}(w_i, w_j) = \left(\sum_{x \in N_j} S^{\theta}(w_i, x) + \sum_{y \in N_i} S^{\theta}(w_j, y) \right)^{\frac{1}{\theta}},$$

$$E_n^{\theta=2}(\text{“}forest\text{”}, \text{“}fruit\text{”}) = \sqrt{(0.10^2 + 0.30^2 + \cdots + 0.01^2) + (0.002^2 + 0^2 + \cdots + 0^2)}$$

(c) sum of squared neighborhood similarities

- Πόσο αυξάνεται η μειώνεται κάθε similarity αν αλλάξουμε την παράμετρο N (μέγεθος της γειτονιάς);
- Επίσης να παρουσιάσετε με όποιον τρόπο θέλετε μερικές γειτονιές λέξεων για N δική σας επιλογής.

Παραδοτέο:

Η εργασία μπορεί να εκπονηθεί με τις ίδιες ομάδες που έχετε δημιουργήσει στην ασκηση 1. Αν πρόκειται για ατομική εργασία, συνεχίζετε ατομικά. .

Θα ανεβάσετε στο eclass ένα φάκελο της μορφής **sdixxxx_sdixxxx.zip** (όπου sdi τα ΑΜ των ατόμων της ομάδας. Αν πρόκειται για ατομική αρκεί ένα sdi στο όνομα του zip αρχείου) Το αρχείο zip πρέπει να περιέχει **ΥΠΟΧΡΕΩΤΙΚΑ** ένα **Python notebook** με το οποίο θα μπορεί κάποιος να δει την εργασία σας βήμα-βήμα. (δηλαδή δεν θα παραδώσετε εκ νέου τα δεδομένα εκπαίδευσης/δοκιμής).

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. Το notebook πρέπει να παραδοθεί “τρεγμένο” με τα αποτελέσματα εμφανή.

Επίσης για όλα τα ερωτήματα (στο μέτρο του εφικτού-χρήσιμου), τα ποιοτικά σχόλια είναι απαραίτητα σχετικά με το τι παρατηρείτε, ποιό συμπέρασμα προκύπτει από τις γραφικές κλπ. Διευκρινίσεις για την εργασία θα δίνονται μέσω του e-class.