NLP 053 - CSE UOI 2025

Ελευθέριος Τράμπας

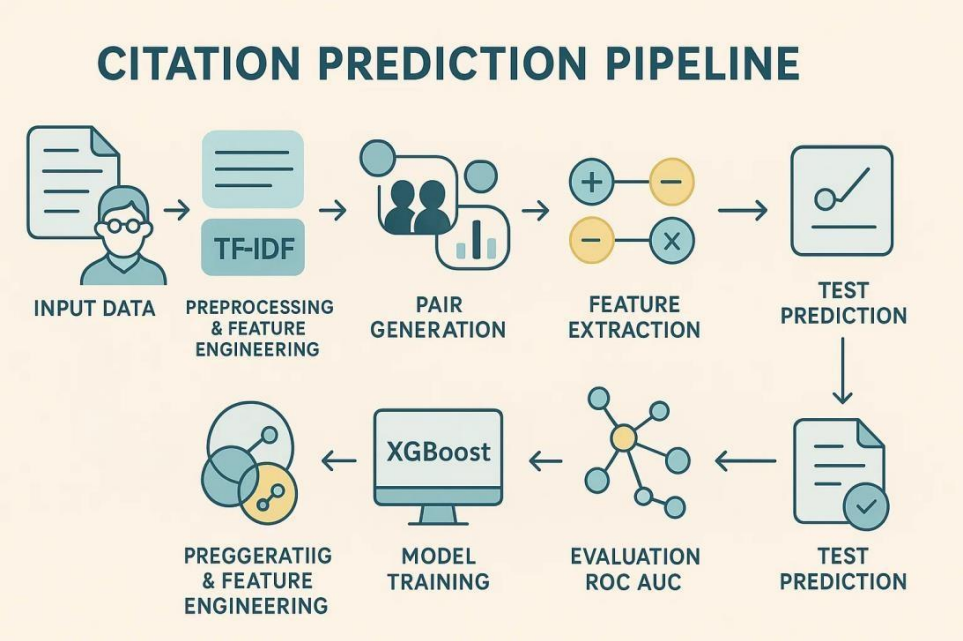
Kaggle :
TRAMPAS ELEFTHERIOS

Εισαγωγή

Το συγκεκριμένο project υλοποιεί ένα μοντέλο το οποίο εκπαιδεύεται στην πρόβλεψη πιθανοτήτων, για το αν ένα paper αναφέρεται σε ένα άλλο. Το πρόβλημα αυτό μπορεί να χαρακτηρηστεί ως ένα πρόβλημα δυαδικής ταξονόμησης, όπου ένα ζεύγος χαρακτηρίζεται ως θετικό (όταν υπάχει αναφορά) και ως αρνητικό (όταν δεν υπάρχει). Για την επίλυση του προβλήματος, υλοποιήθηκε ένα pipeline εξαγωγής χαρακτηριστικών που συνδυάζει διάφορες τεχνικές που σχετίζονται με την φυσική γλώσσα, της θεωρία γράφων και την επεξεργασία κειμένου. Πιο συγκεκριμένα, χρησιμοποιήθηκαν:

- Semantic Embeddings μέσω Sentence Transformers για την αναπαράσταση περιεχομένου των εγγράφων.
- TF-IDF και Jaccard Similarity για την ανάλυση λεξιλογικής ομοιότητας.
- Graph-based χαρακτηριστικά όπως κοινοί γείτονες, διαφορές PageRank και κεντρικότητας.
- Jaccard Similarity μεταξύ συνόλων συγγραφέων για ανίχνευση θεματικής εγγύτητας.
- Διαφορές μήκους tokens για πρόσθετη πληροφορία πολυπλοκότητας κειμένου.

Η τελική συνέννωση αυτών των χαρακτηριστικών οδηγεί στην διαμορφοποίηση ενός εννιαίου διανύσματος το οποίο χρησιμοποιείται ως είσοδος στα μοντέλα (Catboost, XGBoost).



Εικόνα 1. Περιγραφή όλης της διαδικασίας

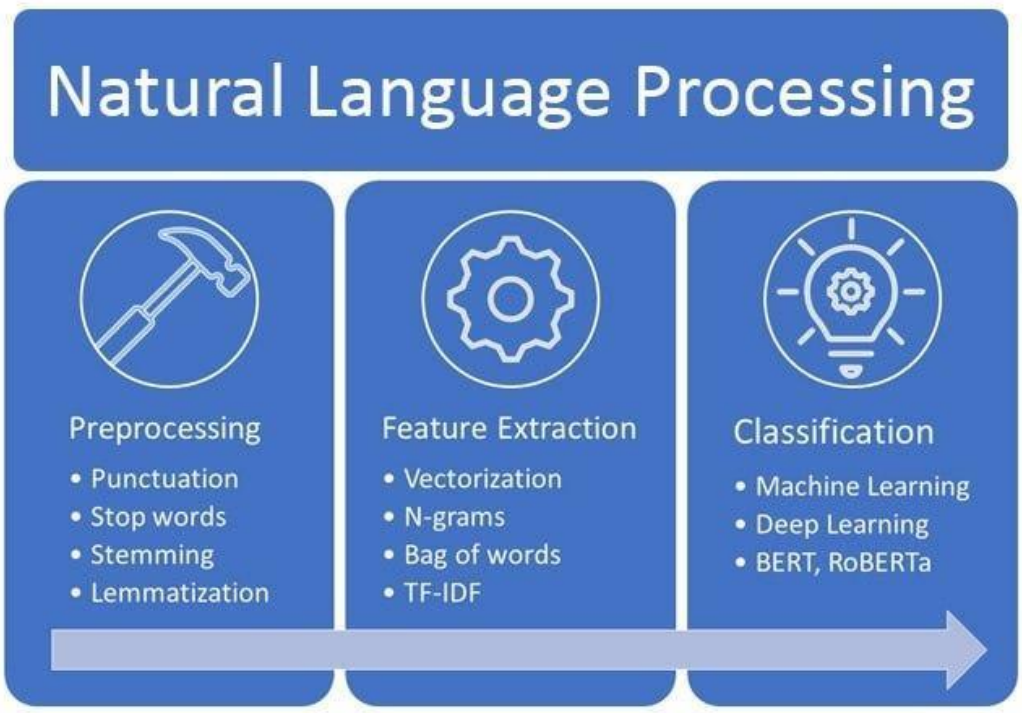
Preprocessing :

Στην προεπεξεργασία των δεδομένων αρχικά φορτώσαμε και συγχωνεύσαμε τα abstracts και τους συγγραφείς με βάση το ID, εξασφαλίζοντας ότι κάθε paper συνοδεύεται από κείμενο και λίστα συγγραφέων. Για το κείμενο εφαρμόσαμε απλό cleaning: μετατροπή σε πεζά, regex που κρατά μόνο λέξεις τουλάχιστον τριών χαρακτήρων, αφαίρεση κενών τιμών ώστε να μην προκαλούνται σφάλματα. Δεν εφαρμόσαμε ρητά αφαίρεση stopwords ή lemmatization, αλλά περιορίσαμε τη διάσταση με TF-IDF (uni- και bi-grams, max_features=5000) και διατηρήσαμε για κάθε έγγραφο το σετ των top-k tokens για υπολογισμό Jaccard similarity. Επιπλέον δημιουργήσαμε embeddings προτάσεων με SentenceTransformer ('all-MiniLM-L6-v2') και CLS token από Roberta-base (truncation σε 128 tokens), στέλνοντας τη φόρτωση στη GPU για ταχύτητα. Για τον γράφο, φορτώσαμε το edgelist σε NetworkX, φτιάξαμε λεξικό id2idx για γρήγορα mapping, υπολογίσαμε για κάθε κόμβο neighbors, degree, PageRank και clustering coefficient, αποθηκεύοντας τιμές με default 0 αν χρειάζεται. Δημιουργήσαμε θετικά ζεύγη από υπάρχουσες ακμές και ισόριθμα αρνητικά τυχαία ζεύγη χωρίς ακμή, διατηρώντας αναπαραγωγικότητα με seed. Κατόπιν, για κάθε ζεύγος εξαγάγαμε πλήθος features: ομοιότητες/αποστάσεις από embeddings, TF-IDF similarities, Jaccard σε top-k tokens και συγγραφείς, καθώς και graph-based δείκτες (διαφορές βαθμών, κοινό πλήθος γειτόνων, Adamic-Adar, διαφορές PageRank και clustering). Μετά εφαρμόσαμε StandardScaler, αν και ιδανικά το fit θα γινόταν μόνο στο training split για αποφυγή data leakage. Το dataset χωρίστηκε stratify σε train/validation.

Feature engineering :

Στο συγκεκριμένο κομμάτι του κώδικα η διαδικασία feature engineering στοχεύει στο να μετατρέψει τα ζεύγη εγγράφων (u, v) σε ένα διάνυσμα χαρακτηριστικών που περιγράφει τη σχέση τους. Για κάθε ζεύγος, αρχικά υπολογίζονται οι ομοιότητες και αποστάσεις στα embeddings κειμένου, τόσο από το SentenceTransformer (sim, dist) όσο και από το Roberta (sim_roberta, cos_roberta, dist_roberta). Αυτά τα χαρακτηριστικά προσφέρουν δεδομένα για το πόσο κοντά αλφαβητικά ή νοηματικά βρίσκονται τα abstracts των δύο κόμβων. Στη συνέχεια αξιοποιούνται TF-IDF διανύσματα για τον υπολογισμό tfidf_sim και Jaccard πάνω στα top-k tokens jacc_tfidf, ώστε να μετρηθεί η επιφανειακή επικάλυψη όρων. Παράλληλα, η ομοιότητα συγγραφέων auth_sim υπολογίζεται επίσης με Jaccard, δίνοντας ένδειξη κοινής συνεργασίας. Στα γραφικά χαρακτηριστικά μετρούνται η απόλυτη διαφορά βαθμού deg_diff, ο αριθμός κοινών γειτόνων cn και το άθροισμα Adamic-Adar aa_score για αυτούς, καθώς και η απόλυτη διαφορά PageRank pr_diff και συντελεστή συσταδοποίησης clust_diff, παρέχοντας πληροφορίες τοπολογίας στο δίκτυο. Τέλος, η διαφορά μήκους κειμένου tok_diff μετράει πόσο διαφέρουν οι

περιλήψεις σε μέγεθος. Όλα αυτά συλλέγονται σε μία λίστα και μετατρέπονται σε numpy array, ώστε να τροφοδοτήσουν το μοντέλο ταξινόμησης. Το αποτέλεσμα είναι ένα ομοιογενές σύνολο χαρακτηριστικών που συνδυάζει στοιχεία νοήματος κειμένου, επιφανειακής αντιστοιχίας όρων, συνεργασίας συγγραφέων και τοπολογικής δομής.

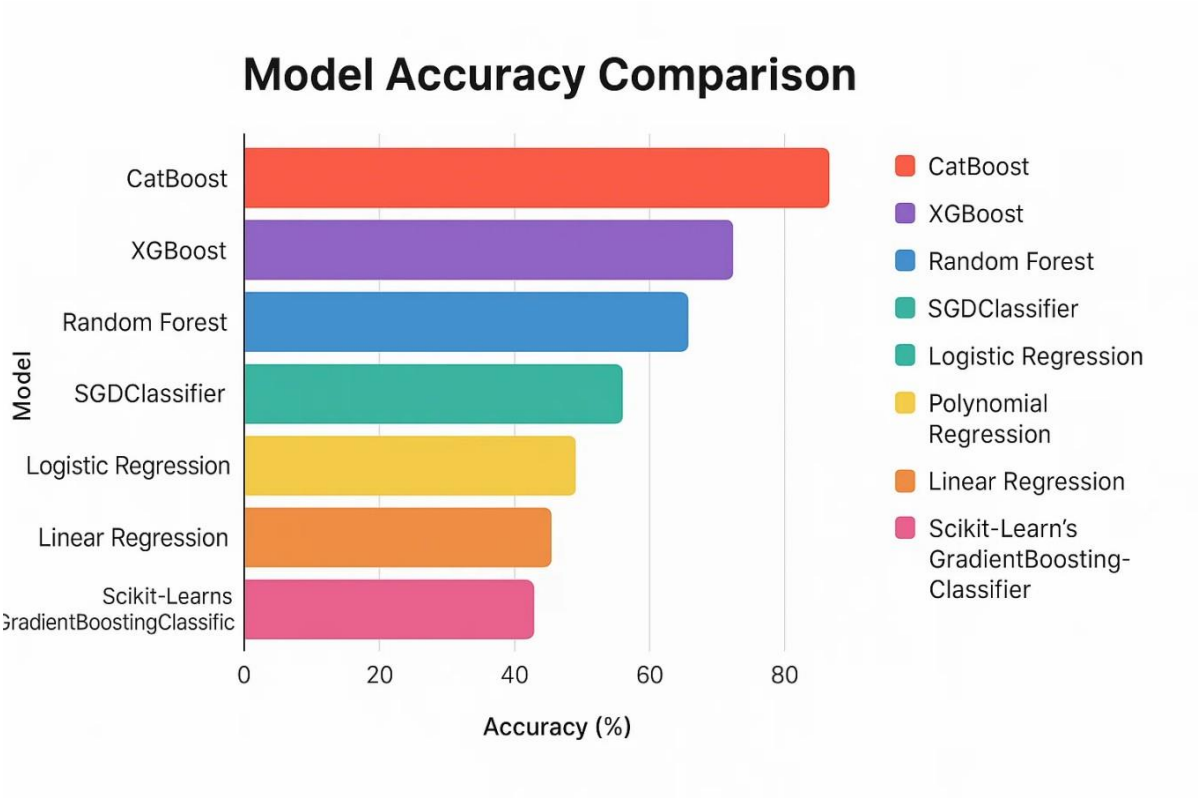


Εικόνα 2. Περιγραφή της διαδικασίας πριν την επιλογή μοντέλου

Models, tuning, and comparison :

Στο συγκεκριμένο τμήμα του κώδικα υλοποιούνται δύο gradient boosting μοντέλα. Αρχικά ορίζεται ένα XGBClassifier με παραμέτρους όπως max_depth=7, learning_rate=0.05, n_estimators=400, subsample=0.8 και colsample_bytree=0.8, οι οποίες έχουν προεπιλεγεί εκ των προτέρων και χρησιμοποιούνται απευθείας στην κλήση fit πάνω στα δεδομένα εκπαίδευσης. Έπειτα ορίζεται ένα CatBoostClassifier με iterations=1000, learning_rate=0.05, depth=8, loss_function='Logloss' και eval_metric='Accuracy', μαζί με early stopping τύπου Iter (od_type='Iter' και od_wait=50) και χρήση GPU (task_type='GPU'). Τα δύο μοντέλα εκπαιδεύονται

ξεχωριστά στα ίδια X_tr, y_tr. Η σύγκριση των μοντέλων γίνεται στο validation set, με τον υπολογισμό Validation AUC.



Other observations:

Παρατηρήσαμε και κάποιες άλλες τεχνικές, οι οποίες συμβάλλουν στην απόδοση του μοντέλου και της ακρίβειας του. Μία απ' αυτές, είναι η μείωση διαστάσεων PCA, όπου μειώνει τη διάσταση των δεδομένων προβλέποντας νέα ορθοκανονικά χαρακτηριστικά που εξηγούν τη μέγιστη διακύμανση. Επιπλέον, απλοποιεί τα πολύπλοκα σύνολα δεδομένων, μειώνει τον θόρυβο και βοηθάει στην οπτικοποίηση υψηλών διαστάσεων.

Ακόμη μια ενδιαφέρον τεχνική είναι και τα SVM's (ή Supported Vector Machines) σε συνδυασμό με τεχνικές SGD. Κύριος στόχος τους, είναι η εύρεση του υπερεπιπέδου που μεγιστοποιεί το περιθώριο ανάμεσα σε κλάσεις .