ΗΜΕΡΟΜΗΝΙΑ: 30/5/2025

NLP Challenge 2025 – Τελική Αναφορά

Ανάλυση, Υλοποιήσεις & Αποτελέσματα Πειραμάτων Ομοιότητας Άρθρων

Περιεχόμενα

1. Εισαγωγή

Αυτή η εργασία έγινε στο πλαίσιο του NLP Challenge με στόχο την πρόβλεψη της ομοιότητας μεταξύ επιστημονικών άρθρων, χρησιμοποιώντας μετα-πληροφορίες (abstracts, authors, citation network). Η πρόβλεψη υποβλήθηκε στην πλατφόρμα Kaggle, και η αξιολόγηση έγινε με βάση το score της στήλης “Similarity” στο test set.

Η μεθοδολογία που ακολουθήθηκε περιελάβανε πολυεπίπεδο pipeline αρχιτεκτονικής, το οποίο περιλαμβάνει την προεπεξεργασία των δεδομένων, εξαγωγή χαρακτηριστικών, μηχανική μάθηση και τελικό μοντέλο με ensembling.

2. Περιγραφή Δεδομένων

Το dataset αποτελείται από τα παρακάτω αρχεία:

- **abstracts.txt**: περιλαμβάνει τα abstracts όλων των άρθρων με μοναδικό ID.
- **authors.txt**: περιλαμβάνει τους συγγραφείς ανά paper.
- **edgelist.txt**: ζεύγη άρθρων που συνδέονται με citation.
- **test.txt**: ζεύγη paper IDs προς πρόβλεψη similarity.

Όλα τα αρχεία καθαρίστηκαν με έλεγχο τύπων, NaN και μη έγκυρων IDs.

3.Preprocessing:

Κατά το στάδιο της προεπεξεργασίας:

- Καθαρισμός και ενοποίηση:
  Ελέγχθηκε η εγκυρότητα των IDs,έγινε η αφαίρεση από κενές ή μη έγκυρες εγγραφές και τέλος έγινε μετατροπή σε κατάλληλους τύπους (π.χ. του uint32).
- Υπολογίσαμε πολλαπλές αναπαραστάσεις των abstracts:
  - **SBERT (BGE)**: μετατρέπει κείμενα σε πολυδιάστατα νοηματικά embeddings βασισμένα σε transformer μοντέλα. Είναι κατάλληλα για γενική κατανόηση του νοήματος.
  - **SPECTER**: ειδικευμένο embedding για επιστημονικά έγγραφα, αποδίδει καλύτερα σε citations.
  - **MPNet & MiniLM**: είναι μοντέλα πιο ελαφριά και γρήγορα, χρήσιμα όταν υπάρχουν περιορισμοί πόρων.
  - **TF-IDF**: μετρά πόσο σημαντική είναι μια λέξη σε ένα κείμενο σε σχέση με ένα σύνολο κειμένων.

- **LDA (Latent Dirichlet Allocation)**: εξάγει θεματικά topics από abstracts για εννοιολογική σύγκριση.
- **Doc2Vec**: βασίζεται στην ιδέα του Word2Vec αλλά εφαρμόζεται σε ολόκληρα κείμενα.

- Υπολογίσαμε **Node2Vec embeddings** για να αποτυπώσουμε το δομικό περιβάλλον κάθε άρθρου στο citation graph.
- Κατασκευάσαμε citation γράφο με NetworkX:
  - **PageRank**: μετρά τη σημασία ενός paper ανάλογα με το πόσα άλλα papers το αναφέρουν, όσο πιο πολλά και σημαντικά, τόσο υψηλότερη η τιμή του
  - **Degree Centrality**: πόσες συνδέσεις έχει ένα paper ,μέτρο δημοφιλίας.
  - **Core Number**: δείχνει πόσο «εσωτερικός» είναι ένας κόμβος στο γράφο.

4. Feature engineering:

Για κάθε ζεύγος papers που εξετάζεται, κατασκευάζονται **πολυδιάστατα χαρακτηριστικά** που προέρχονται από τέσσερις βασικές πηγές: **κείμενο, γράφο, συγγραφείς** και **συνδυαστικά/βοηθητικά γνωρίσματα.**

Textual Similarity

Μετρά πόσο «νοηματικά κοντά» είναι δύο abstracts με βάση τα embeddings τους:

- .Cosine(SBERT), Cosine(SPECTER), Cosine(MiniLM / MPNet): μετρούν την ομοιότητα μεταξύ των πολυδιάστατων αναπαραστάσεων των κειμένων. Όσο πιο κοντά είναι τα διανύσματα, τόσο πιο σχετικά είναι τα κείμενα.
- Cosine(TF-IDF + SVD): συγκρίνει άρθρα ως προς την παρουσία όρων σε μειωμένο χώρο.
- Cosine(LDA): συγκρίνει abstracts με βάση τα «θέματα» στα οποία ανήκουν, π.χ. αν δύο papers σχετίζονται με machine learning.
- Euclidean Distance: χρησιμοποιείται σε Word2Vec / fastText και μετρά «γεωμετρική» απόσταση, όπου μικρότερη σημαίνει μεγαλύτερη ομοιότητα.

Graph-based Features

Αντλούνται από το citation network των άρθρων και περιγράφουν τη **δομή** της σχέσης τους:

- Node2Vec similarity: Μετράει τη δομική ομοιότητα ανάμεσα σε δύο papers μέσα στο citation graph. Δηλαδή, αν τα δύο άρθρα "παίζουν παρόμοιο ρόλο" στο δίκτυο των παραπομπών για παράδειγμα αναφέρονται από παρόμοια papers ή παραπέμπουν σε παρόμοια τότε θεωρούνται πιο σχετικά μεταξύ τους.
- Common Neighbors: πόσοι κόμβοι είναι κοινοί γείτονες των δύο papers, μέτρο τοπικής εγγύτητας.
- Jaccard, Adamic-Adar, Resource Allocation, Preferential Attachment: μετρικές προβλεψιμότητας σύνδεσης στον γράφο.
  - Jaccard: συγκρίνει το πόσους κοινούς γείτονες έχουν δύο κόμβοι, σε σχέση με το σύνολο των γειτόνων τους.
  - Adamic-Adar**:** δίνει μεγαλύτερη σημασία στους κοινούς γείτονες που είναι πιο σπάνιοι.

    - Resource Allocation: προσομοιώνει τη μεταφορά πληροφορίας μεταξύ δύο κόμβων μέσω των κοινών τους γειτόνων.
    - Preferential Attachment: υποθέτει ότι οι κόμβοι με πολλές συνδέσεις έχουν μεγαλύτερη πιθανότητα να αποκτήσουν νέες.
- Shortest Path: μήκος της συντομότερης διαδρομής ανάμεσα στα δύο άρθρα στο γράφο ,μικρότερη απόσταση δείχνει ισχυρότερη σχέση.

Graph Stats

Περιγράφουν τη «σημαντικότητα» των papers μέσα στο δίκτυο:

- PageRank(u,v): πόσο επιρροή έχουν τα δύο papers στο δίκτυο.
- Degree(u,v): πόσα papers συνδέονται άμεσα με τα δύο υπό εξέταση.
- Core Number(u,v): δείχνει σε πόσο «συμπαγή» περιοχή του γράφου ανήκει κάθε paper.

Author-based Feature

Χαρακτηριστικά βασισμένα στους συγγραφείς των άρθρων:

- Jaro-Winkler Author Similarity: μετρά την ομοιότητα δύο λιστών ονομάτων συγγραφέων με fuzzy string matching.
- Fuzzy Jaccard: αναλογικό μέτρο ομοιότητας με fuzzy matching.

Additional

- Citation Flag: boolean που δείχνει αν υπάρχει citation μεταξύ δύο άρθρων (από το edgelist).
- Heuristic Score: χειροποίητος συνδυασμός cosine + jaccard + graph features για γρήγορη πρόβλεψη.

5. Models, tuning, and comparison

Εκπαιδεύσαμε ποικιλία από μοντέλα με πραγματικά labels (0/1):

- RandomForestClassifier / Regressor: ισχυρό, μη γραμμικό μοντέλο με πολλά δέντρα.
- XGBoost / LightGBM: Δύο δημοφιλείς υλοποιήσεις gradient boosting που συνδυάζουν γρήγορη εκπαίδευση, ενσωματωμένο regularization και δυνατότητα early stopping για αποφυγή υπερπροσαρμογής.
- LinearRegression: Απλή μέθοδος παλινδρόμησης που εφαρμόστηκε όταν τα labels αντιμετωπίστηκαν ως συνεχείς τιμές similarity (0–1). Λειτουργεί ως baseline.
- LogisticRegression: Γραμμικό μοντέλο για δυαδική ταξινόμηση. Χρησιμοποιήθηκε επίσης ως meta-learner στο τελικό ensembling.

- CalibratedClassifierCV: βελτιώνει την αξιοπιστία των πιθανοτήτων των προβλέψεων.
- GAT (Graph Attention Network): Μοντέρνο deep learning μοντέλο που αξιοποιεί τη δομή του γράφου. Το GAT λαμβάνει υπόψη του τη σημασία κάθε γείτονα στον γράφο μέσω μηχανισμού attention.

Tuning και αξιολόγηση:

- GridSearchCV: συστηματική αναζήτηση για τις καλύτερες υπερπαραμέτρους.
- StratifiedKFold (5-fold): διασφαλίζει δίκαιη κατανομή θετικών και αρνητικών δειγμάτων.
- Χρησιμοποιήθηκαν μετρικές όπως:

  roc_auc_score: μέτρο διαχωριστικής ικανότητας.

  log_loss: ποινή για λανθασμένες προβλέψεις πιθανοτήτων.

  MAE, RMSE, MSE: απόλυτα και τετραγωνικά σφάλματα για regression μοντέλα.

Ensembling:

- Stacking ensemble: Συνδυάστηκαν οι προβλέψεις από XGBoost, LightGBM και Random Forest με τελικό μοντέλο (meta-learner) την Logistic Regression.
- Rule-based models: Απλές heuristic υλοποιήσεις (cosine + author + pagerank) χρησιμοποιήθηκαν αρχικά για συγκριτικούς σκοπούς και ως baseline.

6. Συμπεράσματα

Ο συνδυασμός διαφορετικών τύπων χαρακτηριστικών ,όπως κειμενική ομοιότητα, γραφο-δομικά μέτρα και πληροφορίες συγγραφέων συνέβαλε ουσιαστικά στη βελτίωση της ακρίβειας των προβλέψεων. Επιπλέον, η χρήση ensembling τεχνικών με meta-learner (Logistic Regression) ενίσχυσε τη γενίκευση του μοντέλου στο test set. Τέλος, η προσεκτική προεπεξεργασία των δεδομένων, όπως η αποθήκευση embeddings σε cache και η πρόβλεψη fallback επιλογών για τα transformer μοντέλα, προσέφερε μεγαλύτερη ταχύτητα και σταθερότητα στην υλοποίηση.

7. Πίνακας Χαρακτηριστικών ανά Pipeline

Παρακάτω παρουσιάζεται ο συγκεντρωτικός πίνακας που συνοψίζει ποια χαρακτηριστικά (feature types) και μοντέλα χρησιμοποιούνται σε κάθε pipeline που αναπτύχθηκε:

Pipeline / Κώδικας	Textual Features	Graph Features	Author Features	Μοντέλα
sbert.py	SBERT	-	Fuzzy Jaccard	Rule-based μόνο
sbert_specter.py	SBERT, SPECTER	-	-	Rule-based
fusion.py	SBERT, MPNet, MiniLM	Node2Vec, Graph Stats	Fuzzy Jaccard	XGB + LGBM + Calibration
0.508.py	SBERT	PageRank, Degree	Fuzzy Jaccard	Rule-based heuristic
prog.py	TF-IDF	Citation flag	Authors	Rule-based heuristic
wdvec2.py	Word2Vec, Doc2Vec	-	-	XGBoostClassifier
tf-dvec.py	TF-IDF, Doc2Vec	-	-	Cosine similarity only
citation_ensemble.py	SBERT, SPECTER, LSI, LDA	Node2Vec, Graph	Jaro-Winkler	LGBM + XGB + RF + LogisticRegression ensemble
improved_pipeline.py	SBERT, SPECTER, LSI, LDA	Node2Vec, Graph Stats	Jaro-Winkler	LGBM + XGB + RF + Meta Learning
0.6045.py	SBERT	PageRank	Fuzzy Jaccard	Linear Regression

8. Ανάλυση Σταδίων Υλοποίησης και Βελτιστοποίησης Pipeline

8.1 Αρχικές Υλοποιήσεις – Στάδιο Πειραματισμού

Στο πρώτο στάδιο , υλοποιήθηκαν αρκετοί πειραματισμοί βασισμένοι κυρίως σε rule-based μεθόδους χωρίς εκπαίδευση μοντέλων. Ο στόχος ήταν να διερευνηθεί η συμβολή διαφορετικών τεχνικών αναπαράστασης (SBERT, TF-IDF, Word2Vec, FastText) στην πρόβλεψη ομοιότητας μεταξύ άρθρων. Οι υλοποιήσεις αυτές εστίασαν στον καθαρισμό δεδομένων, υπολογισμό embeddings και χειροκίνητους κανόνες για την παραγωγή similarity scores.

Ο παρακάτω πίνακας συνοψίζει τα βασικά χαρακτηριστικά των αρχικών scripts:

Αρχείο	Preprocessing	Feature Engineering	Models & tuning
sbert_specter.py	Καθαρισμός abstracts, χρήση MinMaxScaler	SBERT + SPECTER embeddings, cosine sim, ensembling	Όχι μοντέλο, μόνο heuristic similarity
sbert.py	Καθαρισμός, κανονικοποίηση sim [0,1]	SBERT embedding, cosine similarity	Χωρίς classifier, rule-based
progwvec.py	Tokenization + NaN handling	Word2Vec, avg vector, cosine sim	Χωρίς training, rule-based score
progfasttext.py	Μείωση διαστάσεων fasttext + καθαρισμός	fasttext embeddings, cosine sim	Rule-based output, no model
progeucl.py	Tokenization + Word2Vec	Euclidean distance -> similarity	Όχι supervised model
progtfeucl.py	TF-IDF, NaN handling	TF-IDF + Euclidean dist	Όχι training ή validation
proglsa.py	Word2Vec preprocessing	Embeddings + Euclidean similarity	Δεν περιλαμβάνει supervised μάθηση
progloss.py	TF-IDF με log_loss χρήση	cosine sim ως "πρόβλεψη" + label inference	Υποτυπώδες modeling με threshold
prog.py	TF-IDF, χρήση author dict	Text sim + author boost + citation boost	Heuristic tuning, όχι real model

8.2 Παραδοσιακές Υλοποιήσεις με Word2Vec & TF-IDF

Σε αυτό το στάδιο, εξετάστηκαν εναλλακτικές αναπαραστάσεις κειμένου βασισμένες σε Word2Vec και Doc2Vec embeddings, σε συνδυασμό με παραδοσιακά μοντέλα υπολογισμού ομοιότητας όπως cosine similarity και TF-IDF. Οι υλοποιήσεις αυτές έδωσαν έμφαση στην αξιοποίηση συνδυαστικών μεθόδων (π.χ. TF-IDF * Word2Vec) και ελαφριών classifiers όπως XGBoost. Παρότι αρκετές από αυτές βασίζονταν σε rule-based προσέγγιση, βοήθησαν σημαντικά στην κατανόηση της συμπεριφοράς των χαρακτηριστικών και αποτέλεσαν βάση για μετέπειτα βελτιστοποιήσεις.

Αρχείο	Preprocessing	Feature Engineering	Models& tuning
wdvec2.py	Tokenization, PCA, MinMaxScaler	Word2Vec + Doc2Vec embeddings + 3 similarity metrics	XGBoost Classifier με train/test split
wdvec.py	Tokenization + embedding συνδυασμός	Word2Vec + Doc2Vec combined embeddings	Μόνο cosine similarity, χωρίς training
tf-wvec.py	Tokenization + TF-IDF	Word2Vec + TF-IDF ensemble similarity	Χωρίς supervised μοντέλο, rule-based ensembling
tf-dvec.py	Tokenization, TF-IDF, tagged data	Doc2Vec * TF-IDF weighted embedding	Μόνο cosine similarity, χωρίς supervised μοντέλο

8.3 Σύνθετες Υλοποιήσεις – Ενσωμάτωση Μοντέλων & Βελτιστοποίηση

Μετά τις αρχικές πειραματικές προσεγγίσεις, προχωρήσαμε **στη δεύτερη φάση** όπου στόχος ήταν η αξιοποίηση περισσότερων χαρακτηριστικών, η ενσωμάτωση graph-based πληροφορίας και η εφαρμογή μοντέλων μηχανικής μάθησης. Οι παρακάτω υλοποιήσεις βασίστηκαν σε πλήρη προεπεξεργασία, χρήση advanced embeddings και συνδυαστικά χαρακτηριστικά (textual, graph, author). Επιπλέον, εφαρμόστηκε εκπαίδευση μοντέλων όπως XGBoost, LightGBM και Random Forest με ή χωρίς tuning. Ο πίνακας αποτυπώνει αυτή την εξέλιξη:

Αρχείο	Preprocessing	Feature Engineering	Models& tuning
0.815.py	πλήρες	SBERT + authors + pagerank	XGBoost χωρίς GridSearch
0.655.py	πλήρες	SBERT + 4 κεντρικότητες + fuzzy authors	Random Forest με cross-validation
0.6045.py	πλήρες	SBERT + fuzzy + pagerank	LinearRegression + scaling
0.553.py	πλήρες	SBERT + pagerank + graph metrics (Adamic, RA, CN)	XGBoost + GridSearchCV
0.566.py	πλήρες	SBERT + Jaccard + citations	Heuristic rule-based
0.5089.py	πλήρες	SBERT + fuzzy + graph info	Rule-based score
0.508.py	πλήρες	SBERT + fuzzy + pagerank	Χωρίς μοντέλο
0.50.py	πλήρες	SBERT + fuzzy authors + pagerank	Χωρίς training
0.61.py	πλήρες	SBERT + authors/citations boost	Rule-based (no model)

Το **Preprocessing** αναφέρεται ως "πλήρες" , δηλαδή ο κώδικας υλοποιεί όλες τις βασικές και απαιτούμενες ενέργειες προεπεξεργασίας για να μπορέσει το σύστημα να διαβάσει τα δεδομένα και να εκπαιδευτεί σωστά. Πιο συγκεκριμένα, περιλαμβάνει:

Στοιχείο	Περιγραφή
Καθαρισμός abstracts.txt	Αφαίρεση NaN, μετατροπή ID σε uint32, κανονικοποίηση του κειμένου
Καθαρισμός authors.txt	Explode λιστών συγγραφέων, μετατροπή σε set/dict
Καθαρισμός edgelist.txt	Φιλτράρισμα μη έγκυρων IDs, μετατροπή σε uint32, αφαίρεση NaN
Embeddings	Υπολογισμός ή φόρτωση SBERT, BGE, ή SPECTER embeddings
Graph construction (προαιρετικά)	Κατασκευή citation γράφου με NetworkX και υπολογισμός centrality metrics
Ευθυγράμμιση δεδομένων	Φιλτράρισμα test/graph nodes ώστε να αντιστοιχούν μόνο σε valid paper IDs

8.4 Ειδικές Υλοποιήσεις – GAT & Ενοποιημένα Χαρακτηριστικά

Σε αυτό το στάδιο, εστιάσαμε στην εφαρμογή πιο εξειδικευμένων τεχνικών. Συγκεκριμένα:

- Δοκιμάστηκε GAT (Graph Attention Network) για αξιοποίηση του citation graph με attention layers.
- Χρησιμοποιήθηκε ενοποιημένη αναπαράσταση με SBERT + GAT, και αξιολόγηση με MSE Loss.
- Συνδυάστηκαν διαφορετικά embeddings (SBERT, BGE, SPECTER) με graph-based χαρακτηριστικά (για παράδειγμα Jaccard, PageRank).
- Ενσωματώθηκαν πληροφορίες από citations και authors ώστε να ενισχυθεί η ερμηνευσιμότητα και η ακρίβεια.

Οι παρακάτω αντικατοπτρίζουν την προσπάθεια για ενσωμάτωση περισσότερων modal features και τη μετάβαση σε πιο εξειδικευμένα μοντέλα (όπως GNNs, ή stacked regressors).

Αρχείο	Preprocessing	Feature Engineering	Models& tuning
1.22.py	πλήρες	SBERT + GAT + cosine ensembling	GAT με training, απλή MSE
0.73945.py	πλήρες	BGE+SPECTER + pagerank, jaccard	XGBoost + GridSearchCV
0.6553.py	πλήρες	SBERT + Jaccard + pagerank	Random Forest Regressor + CV
0.56374.py	πλήρες	SBERT + cosine + authors/citations	Rule-based score
0.56025.py	πλήρες	SBERT + jaccard + citations	Rule-based, no model

8.5 Τελικές Ενοποιήσεις – Πλήρες Pipeline με Fusion και Calibration

Σε αυτή την τελευταία φάση, σχεδιάστηκαν ολοκληρωμένα pipelines που συνδύασαν όλα τα προηγούμενα επιμέρους χαρακτηριστικά, περιλαμβάνοντας:

- Πλούσια κειμενικά embeddings (BGE, MPNet, MiniLM, SPECTER)
- Χαρακτηριστικά συγγραφέων και γράφου (Jaccard, Node2Vec, PageRank, Common Neighbors, κ.ά.)
- Καθαρή και αποδοτική προεπεξεργασία με caching και modularity
- Εκπαίδευση με XGBoost, LightGBM και Calibration (CalibratedClassifierCV)
- Αξιολόγηση με AUC και visualization feature importances
- Ensembling τεχνικές όπως μέσος όρος calibrated προβλέψεων ή stacking

Αρχείο	Preprocessing	Feature Engineering	Models& tuning
fusion.py	Πλήρης καθαρισμός & caching	BGE + MPNet + MiniLM + Graph + Authors + Node2Vec + engineered features	XGBoost + LightGBM + CalibratedClassifierCV + AUC + ensembling
pipfusion.py	Modular preprocessing με caching	Fusion embeddings + author sim + full graph metrics (PageRank, Jaccard, N2V, κ.ά.)	XGB + LGBM + calibration + ensemble με AUC validation
pippro.py	Φόρτωση, καθαρισμός & Node2Vec (.npz)	Cosine sim από BGE & SPECTER + full graph & author features	XGBoost + GridSearchCV + AUC + normalization + test prediction
pipensemble.py	Επαγγελματικό preprocessing & compatibility	18 χαρακτηριστικά (text, authors, graph, path-based, centralities, core, etc.)	LGBM + XGB + RF + 5-fold CV + stacking ensemble + τελικό blending