Προγραμματιστική εργασία: Σημασιολογική ανάκτηση

Φάση 2

Καραγιαννάκος Άρης ΑΜ 3220066

Κοσμίδη Έλλη ΑΜ 3220086

1+2. Προεπεξεργασία και Δημιουργία Embeddings

Χρήση του `all-MiniLM-L6-v2`.

Η συνάρτηση `encode_texts` με τη βοήθεια της `transformers` βιβλιοθήκης παίρνει τα κείμενα και δημιουργεί embeddings για τα δεδομένα μας. Μέσα στην συνάρτηση πραγματοποιούνται:

1. Batching: Χωρίζουμε τα δεδομένα σε μικρά πακέτα (π.χ. 32 κείμενα τη φορά) για να μην υπερφορτώσουμε τη μνήμη.

2. Model Inference (Forward Pass): Περνάμε τα tokens από το μοντέλο Transformer και παίρνουμε ως έξοδο ένα διάνυσμα για κάθε token της πρότασης.

3. Mean Pooling: Υπολογίζουμε τον μέσο όρο των διανυσμάτων όλων των tokens μιας πρότασης (αγνοώντας τα padding tokens), ώστε να καταλήξουμε σε ένα μοναδικό διάνυσμα (embedding) που αντιπροσωπεύει ολόκληρη την πρόταση.

4. Normalization: Κανονικοποιούμε το τελικό διάνυσμα (L2 norm) ώστε να έχει μήκος 1. Αυτό γίνεται γιατί η ομοιότητα συνημιτόνου (Cosine Similarity) ισοδυναμεί με το εσωτερικό γινόμενο (Dot Product) όταν τα διανύσματα είναι κανονικοποιημένα.

Προεπεξεργασία των δεδομένων:

1. Καθαρισμός (NaN handling)**: Μετατροπή κενών τιμών (NaN) σε κενά strings.

2. Tokenization**: Χωρισμός του κειμένου σε tokens (λέξεις ή τμήματα λέξεων).

3. Mapping**: Αντιστοίχιση των tokens σε μοναδικούς αριθμούς (IDs).

4. Special Tokens**: Προσθήκη ειδικών συμβόλων αρχής [CLS] και τέλους [SEP].

5. Padding**: Προσθήκη "κενών" tokens ώστε όλες οι προτάσεις στο πακέτο να έχουν το ίδιο μήκος.

6. Truncation**: Αποκοπή του κειμένου που ξεπερνά το όριο (128 tokens) για να χωράει στη μνήμη.

7. Tensors**: Μετατροπή των δεδομένων σε μορφή PyTorch Tensors για είσοδο στο μοντέλο.

Στο πρώτο βήμα πραγματοποιήθηκε η φόρτωση και βασική προεπεξεργασία της συλλογής εγγράφων IR2025 (documents.csv) και των ερωτημάτων (queries.csv).

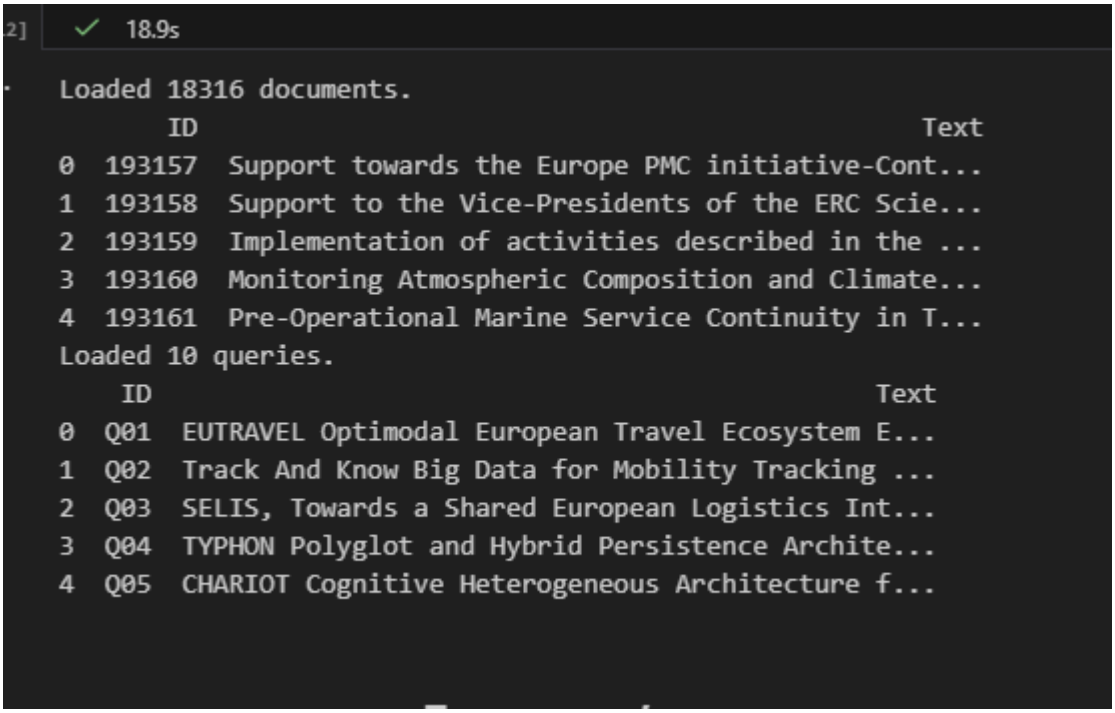
Η προεπεξεργασία περιλάμβανε:

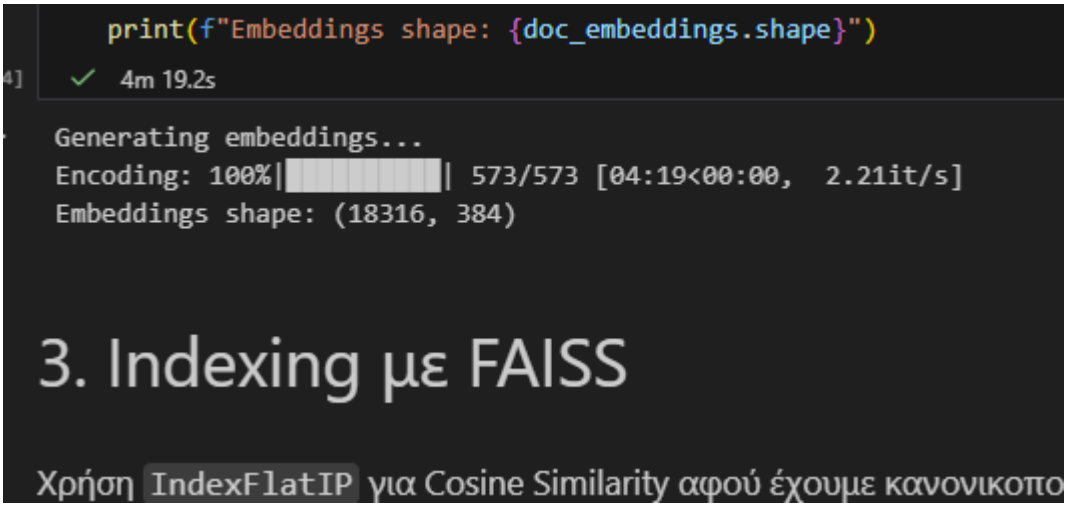
αντικατάσταση τυχόν κενών τιμών (NaN) με κενές συμβολοσειρές,

μετατροπή όλων των κειμένων σε μορφή κατάλληλη για είσοδο σε μοντέλο transformers,

περιορισμό του μήκους των κειμένων στα 128 tokens, ώστε να αποφευχθούν προβλήματα μνήμης και να διατηρηθεί ομοιομορφία στην είσοδο του μοντέλου.

Η παραπάνω διαδικασία εξασφαλίζει ότι όλα τα έγγραφα και τα ερωτήματα μπορούν να επεξεργαστούν με ασφάλεια από το νευρωνικό μοντέλο.





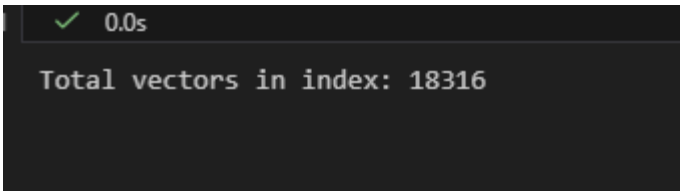
3. Δημιουργία ευρετηρίου FAISS

Στο επόμενο βήμα δημιουργήθηκε ένα ευρετήριο FAISS τύπου IndexFlatIP, το οποίο βασίζεται στο inner product.

Δεδομένου ότι τα embeddings έχουν κανονικοποιηθεί, το inner product ισοδυναμεί με cosine similarity.

Στο ευρετήριο εισήχθησαν όλα τα embeddings των εγγράφων, επιτρέποντας γρήγορη αναζήτηση πλησιέστερων γειτόνων στον πολυδιάστατο διανυσματικό χώρο.

Η χρήση της FAISS επιτρέπει αποδοτική αναζήτηση ακόμα και σε μεγάλες συλλογές, με πολύ χαμηλό υπολογιστικό κόστος σε σύγκριση με εξαντλητικές μεθόδους.



4. Ανάκτηση εγγράφων για κάθε ερώτημα

Για κάθε ερώτημα της συλλογής:υπολογίστηκε το αντίστοιχο embedding με το ίδιο μοντέλο transformers,πραγματοποιήθηκε αναζήτηση στο ευρετήριο FAISS,

ανακτήθηκαν τα k πιο κοντινά έγγραφα για τιμές k = 20, 30 και 50,

τα αποτελέσματα ταξινομήθηκαν κατά φθίνουσα σειρά cosine similarity.

Η διαδικασία επαναλήφθηκε για κάθε τιμή του k, ώστε να μελετηθεί η επίδραση του πλήθους των ανακτηθέντων εγγράφων στην απόδοση του συστήματος.

5. Αξιολόγηση με trec_eval

Για να ετοιμάσουμε τα δεδομένα για το trec_eval:

1. Διασχίσαμε τα αποτελέσματα του FAISS για κάθε ερώτημα.

2. Αντιστοιχίσαμε τους αύξοντες αριθμούς (indices) του FAISS με τα πραγματικά IDs ερωτημάτων και εγγράφων από τα αρχικά CSV αρχεία.

3. Μορφοποιήσαμε κάθε αποτέλεσμα σε μια γραμμή κειμένου με τη δομή: QueryID Q0 DocID Rank Score RunName.

Και εδώ, για λόγους φορητότητας, το path του trec_eval.exe ζητείται δυναμικά από τον χρήστη.

Για κάθε run αρχείο (k=20, 30, 50) υπολογίστηκαν τα ακόλουθα μέτρα:Mean Average,Precision (MAP),Precision@5,Precision@10,Precision@15,Precision@20.

Τα αποτελέσματα συγκεντρώθηκαν σε πίνακα, ο οποίος χρησιμοποιήθηκε για τη σύγκριση των πειραμάτων.

```
=== Phase2 αξιολόγηση για: results_phase2_k20.txt ===
```

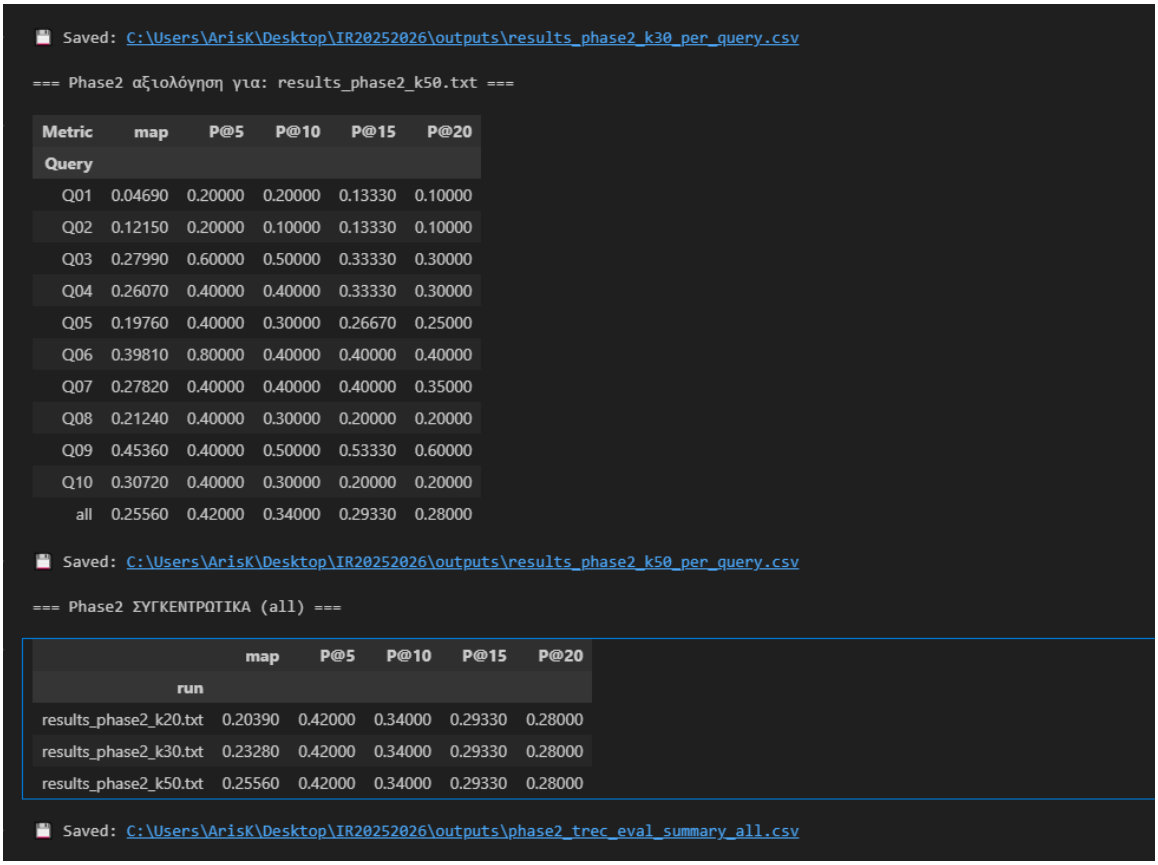
Metric	map	P@5	P@10	P@15	P@20
Query					
Q01	0.03330	0.20000	0.20000	0.13330	0.10000
Q02	0.09850	0.20000	0.10000	0.13330	0.10000
Q03	0.27990	0.60000	0.50000	0.33330	0.30000
Q04	0.22390	0.40000	0.40000	0.33330	0.30000
Q05	0.15520	0.40000	0.30000	0.26670	0.25000
Q06	0.30450	0.80000	0.40000	0.40000	0.40000
Q07	0.21160	0.40000	0.40000	0.40000	0.35000
Q08	0.15250	0.40000	0.30000	0.20000	0.20000
Q09	0.32560	0.40000	0.50000	0.53330	0.60000
Q10	0.25440	0.40000	0.30000	0.20000	0.20000
all	0.20390	0.42000	0.34000	0.29330	0.28000

```
💾 Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\results_phase2_k20_per_query.csv

=== Phase2 αξιολόγηση για: results_phase2_k30.txt ===
```

Metric	map	P@5	P@10	P@15	P@20
Query					
Q01	0.04150	0.20000	0.20000	0.13330	0.10000
Q02	0.12150	0.20000	0.10000	0.13330	0.10000
Q03	0.27990	0.60000	0.50000	0.33330	0.30000
Q04	0.24570	0.40000	0.40000	0.33330	0.30000
Q05	0.18540	0.40000	0.30000	0.26670	0.25000
Q06	0.36530	0.80000	0.40000	0.40000	0.40000
Q07	0.23240	0.40000	0.40000	0.40000	0.35000
Q08	0.18380	0.40000	0.30000	0.20000	0.20000
Q09	0.37970	0.40000	0.50000	0.53330	0.60000
Q10	0.29290	0.40000	0.30000	0.20000	0.20000
all	0.23280	0.42000	0.34000	0.29330	0.28000

```
💾 Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\results_phase2_k30_per_query.csv
```



Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\results_phase2_k30_per_query.csv

=== Phase2 αξιολόγηση για: results_phase2_k50.txt ===

Metric	map	P@5	P@10	P@15	P@20
Query					
Q01	0.04690	0.20000	0.20000	0.13330	0.10000
Q02	0.12150	0.20000	0.10000	0.13330	0.10000
Q03	0.27990	0.60000	0.50000	0.33330	0.30000
Q04	0.26070	0.40000	0.40000	0.33330	0.30000
Q05	0.19760	0.40000	0.30000	0.26670	0.25000
Q06	0.39810	0.80000	0.40000	0.40000	0.40000
Q07	0.27820	0.40000	0.40000	0.40000	0.35000
Q08	0.21240	0.40000	0.30000	0.20000	0.20000
Q09	0.45360	0.40000	0.50000	0.53330	0.60000
Q10	0.30720	0.40000	0.30000	0.20000	0.20000
all	0.25560	0.42000	0.34000	0.29330	0.28000

Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\results_phase2_k50_per_query.csv

=== Phase2 ΣΥΓΚΕΝΤΡΩΤΙΚΑ (all) ===

	map	P@5	P@10	P@15	P@20
run					
results_phase2_k20.txt	0.20390	0.42000	0.34000	0.29330	0.28000
results_phase2_k30.txt	0.23280	0.42000	0.34000	0.29330	0.28000
results_phase2_k50.txt	0.25560	0.42000	0.34000	0.29330	0.28000

Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\phase2_trec_eval_summary_all.csv

6. Καταγραφή πειραμάτων και σύγκριση με Φάση 1

Στο τελευταίο βήμα, τα αποτελέσματα της Φάσης 2 συγκρίθηκαν με εκείνα της Φάσης 1 (λεξιλογική ανάκτηση).

Η σύγκριση έγινε με βάση τις ίδιες μετρικές αξιολόγησης.

Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\results_phase2_k50_per_query.csv

=== Phase2 ΣΥΓΚΕΝΤΡΩΤΙΚΑ (all) ===

run	map	P@5	P@10	P@15	P@20
results_phase2_k20.txt	0.20390	0.42000	0.34000	0.29330	0.28000
results_phase2_k30.txt	0.23280	0.42000	0.34000	0.29330	0.28000
results_phase2_k50.txt	0.25560	0.42000	0.34000	0.29330	0.28000

Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\phase2_trec_eval_summary_all.csv

=== ΣΥΓΚΡΙΣΗ Phase1 vs Phase2 ===

	map	P@5	P@10	P@15	P@20
Phase1_run_bm25_k20.txt	0.54630	0.82000	0.67000	0.58000	0.52500
Phase1_run_bm25_k30.txt	0.60110	0.82000	0.67000	0.58000	0.52500
Phase1_run_bm25_k50.txt	0.63850	0.82000	0.67000	0.58000	0.52500
Phase2_results_phase2_k20.txt	0.20390	0.42000	0.34000	0.29330	0.28000
Phase2_results_phase2_k30.txt	0.23280	0.42000	0.34000	0.29330	0.28000
Phase2_results_phase2_k50.txt	0.25560	0.42000	0.34000	0.29330	0.28000

Saved: C:\Users\ArisK\Desktop\IR20252026\outputs\comparison_phase1_vs_phase2.csv

DONE — Phase2 evaluation ολοκληρώθηκε ✓