ΣΥΣΤΗΜΑΤΑ ΑΝΑΚΤΗΣΗΣ ΠΛΗΡΟΦΟΡΙΩΝ
2η Φάση

Χαράλαμπος Ταράτσας 3220255
Αναστασία Ανδρομίδα 3210008

Το πρώτο βήμα για τη Φάση 2 ήταν η επεξεργασία των αρχείων συνωνύμων που παρέχονται από το WordNet. Τα αρχεία αυτά (π.χ., wn_s.pl, wn_s_verbs.pl) είναι σε μορφή η οποία δεν είναι συμβατή με την Elasticsearch. Για τον λόγο αυτό, αναπτύξαμε το Python script convert_synonyms.py. Με την χρήση του script αυτού δημιουργήσαμε τα αρχεία wordnet_synonyms_converted.txt και wordnet_verb_synonyms_converted.txt. Ενδεικτικά οι πρώτες 5 γραμμές του wordnet_synonyms_converted.txt είναι:

whole,unit

organism,being

person,individual,someone,somebody,mortal,soul

animal,beast,brute,creature,fauna

plant,flora

Αυτή η διαδικασία επαναλήφθηκε για κάθε σενάριο της Φάσης 2, χρησιμοποιώντας το αντίστοιχο αρχείο εισόδου (wn_s.pl για το σενάριο "All Synonyms" και wn_s_verbs.pl για το "Verbs Only") για να παραχθεί το κατάλληλο αρχείο

Έπειτα, το κύριο script phase2_wordnet_elastic.py είναι υπεύθυνο για την επικοινωνία με την Elasticsearch. Η κεντρική του λειτουργία είναι η δημιουργία ενός index με έναν ειδικά διαμορφωμένο analyzer (wordnet_analyzer) που ενσωματώνει τα συνώνυμα.

Ο **wordnet_analyzer** ορίζεται με την εξής δομή:

Tokenizer: Χρησιμοποιείται ο standard tokenizer, ο οποίος χωρίζει το κείμενο σε όρους βάσει των κενών και των σημείων στίξης.

Token Filters: Εφαρμόζονται δύο φίλτρα με τη σειρά:

lowercase: Μετατρέπει όλους τους όρους σε πεζά γράμματα.

wordnet_synonyms_filter: Ένα custom token filter τύπου synonym_graph. Αυτό το φίλτρο είναι που κάνει την επέκταση. Διαβάζει το αρχείο wordnet_synonyms_converted.txt (το οποίο πρέπει να βρίσκεται στον κατάλληλο υποφάκελο config/analysis/ μέσα στον φάκελο της Elasticsearch, στην δική μας περίπτωση πρέπει να το βάλουμε στο σωστό path στο docker container απο το οποιο τρέχουμε την Elasticsearch, καθως η συμβατική εγκατάσταση σε MacOS δημιουργούσε προβλήματα) και αντικαθιστά ή προσθέτει τους συνώνυμους όρους στον ροή του κειμένου.

Τέλος, στο mapping του index, το πεδίο text_content (που περιέχει το κείμενο των εγγράφων) ορίζεται να χρησιμοποιεί τον wordnet_analyzer τόσο κατά την ευρετηρίαση (analyzer) όσο και κατά την αναζήτηση (search_analyzer). Αυτό διασφαλίζει ότι τόσο τα έγγραφα όσο και τα ερωτήματα υφίστανται την ίδια επεξεργασία επέκτασης, αποφεύγοντας αναντιστοιχίες.

Συνεχίζοντας, η συνάρτηση index_ir2025_collection στο phase2_wordnet_elastic.py αναλαμβάνει την ευρετηρίαση των εγγράφων.

Διαβάζει το αρχείο της συλλογής (corpus.jsonl) γραμμή προς γραμμή.

Για κάθε έγγραφο JSON, εξάγει το αναγνωριστικό (doc_id ή paper_id) και το περιεχόμενο.

Για αποδοτικότερη ευρετηρίαση, χρησιμοποιεί τον μηχανισμό bulk της Elasticsearch, ο οποίος επιτρέπει την αποστολή πολλαπλών εγγράφων προς ευρετηρίαση σε μία μόνο κλήση δικτύου.

Τέλος η συνάρτηση run_ir2025_queries_and_collect_results είναι υπεύθυνη για την εκτέλεση των σεναρίων.

Διαβάζει το αρχείο ερωτημάτων (queries.jsonl), το οποίο είναι σε μορφή JSONL.

Για κάθε γραμμή, αναλύει το JSON και εξάγει το _id του ερωτήματος και το κείμενο του ερωτήματος (text).

Στέλνει ένα match query στην Elasticsearch, στοχεύοντας το πεδίο text_content. Επειδή αυτό το πεδίο χρησιμοποιεί τον wordnet_analyzer, η επέκταση του ερωτήματος με συνώνυμα γίνεται αυτόματα από την Elasticsearch κατά τον χρόνο της αναζήτησης.

Μορφοποιεί τα αποτελέσματα σύμφωνα με τις απαιτήσεις του trec_eval (δηλαδή, query_id Q0 doc_id rank score run_id) και τα αποθηκεύει σε ένα αρχείο εξόδου results.txt.

Έτσι, για κάθε σενάριο έχουμε:

Μετρική	Φάση 1	Φάση 2 (WordNet - "All Syns")	Φάση 2 (WordNet - "Verbs Only")
runid	phase1	phase2_wordnet_all_syns	phase2_wordnet_verbs_syns
MAP	0.0211	0.0216	0.0241
P@5	252	256	328
P@10	0.26	266	302
P@15	0.2627	0.2653	0.3027
P@20	254	257	291
num_rel_ret	1056	1061	1084

Συγκρίνοντας το σενάριο επέκτασης με όλα τα συνώνυμα με τα αποτελέσματα της φάσης 1, παρατηρούμε ότι οι αλλαγές στην απόδοση είναι ελάχιστες. Η μετρική MAP παρουσίασε μια οριακή αύξηση από 0.0211 σε 0.0216, ενώ οι μετρικές P@k είχαν επίσης αμελητέα βελτίωση (π.χ., η P@5 αυξήθηκε από 0.2520 σε 0.2560). Το εύρημα αυτό υποδηλώνει ότι η "αφελής" προσθήκη όλων των πιθανών συνωνύμων από το WordNet δεν ήταν αποτελεσματική. Η κύρια αιτία είναι πιθανότατα το φαινόμενο του Query Drift. Η προσθήκη όρων που μπορεί να ανήκουν σε διαφορετικό εννοιολογικό πλαίσιο από αυτό του αρχικού ερωτήματος εισήγαγε "θόρυβο" στην αναζήτηση. Αυτός ο θόρυβος φαίνεται πως αντιστάθμισε οποιοδήποτε πιθανό όφελος από την προσθήκη σωστών και σχετικών συνωνύμων, οδηγώντας σε μια τελικά κακή απόδοση

Σε αντίθεση με το προηγούμενο σενάριο, η στοχευμένη επέκταση μόνο με συνώνυμα ρημάτων έφερε μια **σαφή και στατιστικά σημαντική βελτίωση** σε σχέση με το Baseline.

- Η μετρική **MAP** αυξήθηκε κατά περίπου 14% (από 0.0211 σε 0.0241), υποδεικνύοντας μια συνολικά καλύτερη κατάταξη των σχετικών εγγράφων.
- Η μεγαλύτερη βελτίωση παρατηρήθηκε στην **P@5**, η οποία αυξήθηκε κατά περίπου 30% (από 0.2520 σε 0.3280).
- Αντίστοιχες βελτιώσεις παρατηρήθηκαν και στις μετρικές P@10, P@15 και P@20.

Το συμπέρασμα είναι ότι ο περιορισμός της επέκτασης στα ρήματα μείωσε δραστικά τον θόρυβο που παρατηρήθηκε στο Σενάριο 1. Για τη συγκεκριμένη συλλογή και τα ερωτήματα, τα οποία συχνά εκφράζουν μια διαδικασία (π.χ., "how does the coronavirus respond...", "what causes death..."), η επέκταση των ρημάτων αποδείχθηκε μια αποτελεσματική στρατηγική για την κάλυψη διαφορετικών λεκτικών διατυπώσεων της ίδιας πληροφοριακής ανάγκης.