Συστήματα ανάκτησης πληροφοριών

3η Φάση Προγραμματιστικής Εργασίας

Βιβλιοθήκη LISA

Ον/μο: ΒΙΤΑΛΗΣ ΙΩΑΝΝΗΣ
ΑΜ: 3150011

Δομή

Το παραδοτέο για την 3η Φάση της προγραμματιστικής εργασίας αποτελείται από 2 φακέλους, τον BM25 και τον LMJ, που χρησιμοποιούν τις συναρτήσεις ομοιότητας “BM25Similarity” και “LMJelinekMercerSimilarity” αντίστοιχα.

Κάθε φάκελος περιέχει τον src κώδικα με τις αλλαγές που χρειάστηκαν από την 1η Φάση της προγραμματιστικής εργασίας και έναν φάκελο trec_eval με .txt αρχεία για τα αποτελέσματα που προέκυψαν από την χρήση του εργαλείου αξιολόγησης trec_eval

Διαφορές στον κώδικα

Χρησιμοποίησα τον κώδικα που υπέβαλα για την 1η Φάση της προγραμματιστικής εργασίας με μικρές αλλαγές προκειμένου να χρησιμοποιηθούν οι σωστές συναρτήσεις ομοιότητας.

- **BM25**
  - Στον Indexer.java του package myLuceneApp άλλαξα το Similarity από ClassicSimilarity σε BM25Similarity()
  - Στον Searcher.java του package myLuceneApp άλλαξα το .setSimilarity() του IndexSearcher από ClassicSimilarity σε BM25Similarity
- **LMJ**
  - Στον Indexer.java του package myLuceneApp άλλαξα το Similarity από ClassicSimilarity σε “LMJelinekMercerSimilarity((float) 0.7)”
  - Στον Searcher.java του package myLuceneApp άλλαξα το .setSimilarity() του IndexSearcher από ClassicSimilarity σε “LMJelinekMercerSimilarity((float) 0.7)”

- **Αποτελέσματα BM25 στον φάκελο BM25\\trec_eval**

Μέτρο Αξιολόγησης / **Κ πρώτα Ανακτηθέντα κείμενα**	MAP (Mean Average Precision)	avgPre@k (Average Precision @k)
K = 20	0.2428	P_05 = 0.3100 P_10 = 0.2350 P_15 = 0.1767 P_20 = 0.1525
K = 30	0.2507	P_05 = 0.3100 P_10 = 0.2350 P_15 = 0.1767 P_20 = 0.1525
K = 50	0.2585	P_05 = 0.3100 P_10 = 0.2350 P_15 = 0.1767 P_20 = 0.1525

Παρατηρούμε καλύτερα αποτελέσματα συγκριτικά με την 1η Φάση της προγραμματιστικής εργασίας. Γνωρίζουμε οτι και το Classic Similarity και το BM25 χρησιμοποιούν την ίδια αριθμητική στατιστική για να δείξουν πόσο σημαντική είναι μια λέξη για ένα έγγραφο σε μια συλλογή κειμένων που ονομάζεται TF*IDF (ή αλλιώς term frequency–inverse document frequency) που βασίζεται στο μοντέλο Bag-Of-Words.

Το απλό TF-IDF που χρησιμοποιεί το classic similarity επιβραβεύει τη συχνότητα των όρων και τιμωρεί τη συχνότητα των εγγράφων. Ο BM25 λαμβάνει υπόψη του το μήκος των εγγράφων και τον κορεσμό της συχνότητας των όρων, προσφέροντας συνήθως καλύτερα απότελεσματα ειδικά σε μεγαλύτερα κείμενα και μεγάλες συλλογές κειμένων.

- **Αποτελέσματα LMJ στον φάκελο LMJ\\trec_eval**

Μέτρο Αξιολόγησης / **Κ πρώτα Ανακτηθέντα κείμενα**	MAP (Mean Average Precision)	avgPre@k (Average Precision @k)
K = 20	0.2166	P_05 = 0.2800 P_10 = 0.2150 P_15 = 0.1767 P_20 = 0.1500
K = 30	0.2267	P_05 = 0.2800 P_10 = 0.2150 P_15 = 0.1767 P_20 = 0.1500
K = 50	0.2301	P_05 = 0.2800 P_10 = 0.2150 P_15 = 0.1767 P_20 = 0.1500

Στην περίπτωση του LMJ, η συνάρτηση ομοιότητας δέχεται έναν αριθμό στο πεδίο τιμών [0,1].

- Για μικρά λ, συμβαίνει μια πιο συζευκτική ανάζητηση των όρων, άρα τα έγγραφα που ταιριάζουν με περισσότερους όρους του ερωτήματος θα κατατάσσονται υψηλότερα από εκείνα που ταιριάζουν με λιγότερους όρους. Για τα ερωτήματα μικρού μεγέθους, η απόδοση ανάκτησης τείνει να βελτιστοποιείται όταν το λ είναι μικρό.

- Για μεγάλα λ, συμβαίνει διαζευκτική αναζήτηση των όρων άρα το «βάρος» ενός όρου λαμβάνεται υπόψην, αυτό σημαίνει ότι η κατάταξη των κειμένων εξαρτάται από τον όρο με το μεγαλύτερο βάρος. Για τα μακροσκελή ερωτήματα, υψηλότερες τιμές του λ συνήθως φέρνουν καλύτερα αποτελέσματα.

Ερευνόντας την συλλογή μου παρατήρησα οτι αποτελείται κυρίως από μακροσκελή κείμενα αφού στον Indexer έχω χρησιμοποιήσει και τον τίτλο και το κυρίως κείμενο, άρα μια μεγάλη τιμή στο λ θα προσέφερε καλύτερα αποτελέσματα.

Δοκίμασα τιμές από 0.5 έως 0.8 και κατέληξα οτι η καλύτερη τιμή ήταν το 0.7.