Ανάκτηση και Εξόρυξη Πληροφοριών
Προγραμματιστική Εργασία #1

Διδάσκων:
Χ. Τρυφωνόπουλος

Παράδοση μέχρι: Κυριακή 14/12/2025 ώρα 23.59
Προσωπική εξέταση: στο τέλος του εξαμήνου

ΣΗΜΑΝΤΙΚΕΣ ΣΗΜΕΙΩΣΕΙΣ:

1. Αφού έχετε ολοκληρώσει την άσκηση που θέλετε να παραδώσετε, την υποβάλετε στο eclass στο υποσύστημα «Εργασίες». Η υποβολή πρέπει να γίνει ΠΡΙΝ την ημερομηνία παράδοσης. Παραδίδετε όλα τα απαραίτητα αρχεία σε ένα zip (κώδικας, εκτελέσιμο, συνοδευτικά αρχεία, και αναφορά).

2. Η άσκηση συνίσταται ισχυρά να υλοποιηθεί από ομάδα δύο ατόμων, αλλά μπορεί να υλοποιηθεί και ατομικά. Περιπτώσεις αντιγραφής θα μηδενίζονται και το ζήτημα θα συζητείται στη συνέλευση Τμήματος. Η παράδοση θα πρέπει να γίνει εντός της προθεσμίας και μόνο μέσω του eclass, όχι με email στον διδάσκοντα.

Στην άσκηση αυτή καλείστε να υλοποιήσετε τη μηχανή αναζήτησης **TrASH (Trump Automated Search Hub)**, η οποία θα υποστηρίζει την αποθήκευση, ευρετηρίαση και αναζήτηση των Facebook posts του Προέδρου των ΗΠΑ Donald J. Trump, καθώς είναι κρίμα (#not_really) να χαθεί τόσο γλαφυρός και χειμαρρώδης διαδικτυακός λόγος.

Η υλοποίησή σας θα βασιστεί στο σύστημα ElasticSearch, μια NoSQL βάση δεδομένων που ειδικεύεται στον ευρετηριασμό και την ανάκτηση κειμενικής πληροφορίας και είναι χτισμένη πάνω από το σύστημα Apache Lucene έχοντας όμως, καλύτερη υποστήριξη σε πληροφορίες εκμάθησης (tutorials, παραδείγματα, κ.λπ.) και μία δυναμική κοινότητα χρηστών. Το ElasticSearch προσφέρει βασικές λειτουργίες Ανάκτησης Πληροφοριών, όπως οργάνωση της συλλογής κειμένων με τη δημιουργία ευρετήριων, αναζήτηση κειμένων με ερωτήσεις εκφρασμένες σε διάφορα μοντέλα (όπως Boolean, Vector Space, BM25, φράσεων, κ.λπ.), και κατάταξη των σχετικών αποτελεσμάτων, ενώ τα τελευταία χρόνια γίνεται όλο και πιο δημοφιλές για την ανάπτυξη εφαρμογών που απαιτούν λειτουργίες ανάκτησης κειμένων λόγω και των σημαντικών του επιδόσεων.

Ως γλώσσα υλοποίησης προτείνονται η Java ή/και η Python μιας και υποστηρίζονται από το ElasticSearch, αλλά μπορείτε να κάνετε την υλοποίηση επιμέρους κομματιών ή όλου του project σε όποια γλώσσα επιθυμείτε. Μαζί με τον πηγαίο κώδικα του προγράμματός σας θα παραδώσετε και τυχόν

συνοδευτικά αρχεία, το εκτελέσιμο αρχείο, και μια αναφορά. Έμφαση θα δοθεί στην κομψότητα της υλοποίησης, στην ταχύτητα εκτέλεσης των ζητούμενων, στη χρηστικότητα και ορθότητα του προγράμματός σας, και στην πληρότητα και σαφήνεια των λοιπών παραδοτέων. ΜΗΝ συμπεριλάβετε στην αναφορά σας μέρος ή σύνολο του κώδικα!

Περιγραφή του προβλήματος και των δεδομένων

Στο πρόβλημα που έχετε να αντιμετωπίσετε, σας δίνεται μία μικρή (αλλά γλαφυρή και πλήρως αντιπροσωπευτική) συλλογή από 4.000 post (<1MB ασυμπίεστη) που έχει δημοσιεύσει στο Facebook ο Πρόεδρος των ΗΠΑ και έχουν συλλεχθεί με αυτοματοποιημένες μεθόδους. Κάθε post αποτελείται από τα παρακάτω πεδία:

- το κείμενο του post
- μεταδεδομένα για το πού βρέθηκε ή πού αναφέρεται το post
- ο τύπος του post (αν είναι βίντεο, εικόνα, σύνδεσμος, κ.λπ.)
- τυχόν link που περιέχεται στο post
- η ημερομηνία και ώρα δημοσίευσης του post
- το σύνολο των reactions που έλαβε (likes/loves/wows/hahas/sads/angrys)
- το σύνολο των σχολίων
- το σύνολο των shares
- 6 πεδία με τον αριθμό των likes/loves/wows/hahas/sads/angrys αναλυτικά.

Για τους σκοπούς της άσκησης μας ενδιαφέρουν όλα τα παραπάνω πεδία.

Τα κείμενα για ευκολία θα πρέπει να θεωρήσετε ότι είναι οργανωμένα σε μία συλλογή (document collection), η οποία βρίσκεται αποθηκευμένη σε ένα csv αρχείο στο δίσκο, και θα πρέπει να επιτρέπετε στο χρήστη να προσθέτει, να αφαιρεί, ή να τροποποιεί post στη συλλογή που ευρετηριάζει το TrASH. Τα ζητήματα αυτά περιγράφονται αναλυτικά στις παρακάτω ενότητες.

Λειτουργικότητα της μηχανής αναζήτησης TrASH

Η μηχανή αναζήτησης TrASH θα πρέπει να παρέχει τη λειτουργικότητα που περιγράφεται παρακάτω.

Προεπεξεργασία περιεχομένου posts [10%]

Αρχικά, θα παίρνετε ως είσοδο ένα αρχείο csv, το οποίο περιέχει έναν αριθμό από post με τη δομή που περιγράφηκε στην προηγούμενη ενότητα, και θα πρέπει να προεπεξεργαστείτε το αρχείο αυτό κατάλληλα ανάλογα με τον τύπο του πεδίου. Η διαδικασία αυτή μπορεί να γίνει είτε με κώδικα που θα γράψετε εσείς, είτε με κάποιον parser τρίτου. Εννοείται ότι αν κάνετε χρήση έτοιμου parser θα πρέπει να αναφέρετε την πηγή σας στον κώδικα και στην αναφορά. Κατόπιν σκεφτείτε αν χρειάζεται κάποιο άλλο είδος προεπεξεργασίας, όπως αν θα παραλείψετε τις λέξεις αποκλεισμού (stopwords), αν θα μετατρέψετε τις λέξεις σε μικρά/κεφαλαία γράμματα (case folding), αν θα κάνετε λημματοποίηση (stemming), και αν θα σβήσετε τα σημεία στίξης (punctuation removal), καθώς και σε ποια πεδία θα πρέπει να εφαρμοστεί

η προεπεξεργασία αυτή. Προφανώς δεν θα πρέπει να εφαρμόσετε έναν ενιαίο τύπο προεπεξεργασίας για όλα τα πεδία καθώς οι ανάγκες αναζήτησης είναι διαφορετικές ανά πεδίο (π.χ., αλλιώς ψάχνει κάποιος στο κείμενο ενός post, αλλιώς σε ένα URL ή σε μία ημερομηνία).

Η λειτουργικότητα για την παραπάνω επεξεργασία παρέχεται σε μεγάλο βαθμό από το ElasticSearch, αλλά μπορείτε να χρησιμοποιήσετε και κώδικα δικό σας ή τρίτων, αρκεί να δίνετε την πηγή στη γραπτή αναφορά σας.

Κατασκευή ευρετηρίων και εισαγωγή/διαγραφή posts [30%]

Η μηχανή αναζήτησης TrASH θα πρέπει να παρέχει τη δυνατότητα εισαγωγής και διαγραφής ενός ή πολλών post. Η εισαγωγή θα γίνεται μέσω κατάλληλα διαμορφωμένων αρχείων csv με τη δομή που περιγράφηκε παραπάνω, ενώ η διαγραφή θα πρέπει να υποστηρίζεται με κατάλληλο τρόπο (π.χ., επιλέγοντας ένα ή περισσότερα post και δίνοντας κατάλληλη εντολή διαγραφής από το γραφικό περιβάλλον).

Σημειώστε ότι η εισαγωγή/διαγραφή ενός ή περισσοτέρων εγγράφων θα πρέπει να συνεπάγεται την ενημέρωση των ευρετηρίων που κρατούνται στο ElasticSearch (ή τη δημιουργία τους αν η συλλογή είναι κενή). Η κατασκευή του ευρετηρίου αυτού θα επιτρέψει την εκτέλεση των ερωτημάτων που περιγράφονται στην παρακάτω ενότητα.

Αν δοθεί λάθος όνομα αρχείου εισαγωγής, θα πρέπει να χειρίζεστε το λάθος με κατάλληλο μήνυμα, ενώ δεν χρειάζεται να ελέγχετε αν κάποιο post που εισάγεται υπάρχει ήδη στο ευρετήριο καθώς δεν θα έπρεπε να υπάρχει όριο στο πόσες φορές θα διαβάσει κάποιος τέτοια ~~παραλληρήματα~~ σοφά λόγια.

Αναζήτηση και ομοιότητα post [30%]

Θα πρέπει να υποστηρίζεται η δυνατότητα αναζήτησης post με βάση ένα ή περισσότερα πεδία και να παρέχεται στο χρήστη η δυνατότητα διατύπωσης ερωτημάτων Boolean (τελεστές AND, OR, NOT), Vector Space, και φράσεων για τα πεδία που αποθηκεύουν μη αριθμητική πληροφορία (όπως κείμενο, URL, κ.λπ.). Σημειώστε ότι το ElasticSearch υποστηρίζει εγγενώς όλους τους παραπάνω τύπους ερωτήσεων, επομένως για να υποστηρίξετε τέτοια ερωτήματα θα πρέπει απλώς να καλέσετε την κατάλληλη μέθοδο από αυτές που παρέχει το ElasticSearch. Θα πρέπει επίσης να υποστηρίζετε κατάλληλου τύπου ερωτήσεις (π.χ., μεγαλύτερο από/μικρότερο από/ίσο/πριν από/μετά από) και για τα αριθμητικά πεδία και για τα πεδία ημερομηνίας/ώρας, είτε μέσω του ElasticSearch είτε μέσω δικής σας υλοποίησης.

Σημειώστε ότι θα πρέπει να σκεφτείτε και να αποφασίσετε τον τρόπο με τον οποίο θα επιτρέπετε στο χρήστη να κάνει ερωτήσεις που θέτουν περιορισμούς σε ένα ή παραπάνω πεδία συγχρόνως (π.χ., θέλω να βρω τα post που μιλούν για τη Hilary Clinton και έχουν πάνω από 500 likes, ή θέλω να βρω video post που μιλούν για corruption μετά από τον Οκτώβρη του 2016). Προσπαθήστε να φτιάξετε ένα απλό και φιλικό τρόπο υποβολής ερωτημάτων, ο οποίος παράλληλα θα είναι και εύληπτος από το χρήστη. Αποφασίστε πώς θα υποστηρίξετε τέτοιες σύνθετες ερωτήσεις (π.χ., με ένα ενιαίο πεδίο, με πολλά

που θα συνδυάζονται, με δυναμική προσθήκη πεδίων) και μην ξεχάσετε να έχετε διαθέσιμα στο χρήστη παραδείγματα των υποστηριζόμενων ερωτημάτων.

Εκτός από την αναζήτηση post, το σύστημα TrASH θα πρέπει να υποστηρίζει και την εύρεση post που είναι συναφή με άλλα ήδη αποθηκευμένα. Σε αυτή τη λειτουργία ο χρήστης θα επιλέγει ένα post και θα ζητά από τη μηχανή αναζήτησης TrASH να του φέρει μια λίστα με τα top-K πιο συναφή με αυτό post με βάση το κείμενο (ανάλογα με το K που έδωσε ή έχει θέσει ως προ-ρύθμιση στο σύστημα ο χρήστης).

Γραφικό περιβάλλον χρήσης και προβολή αποτελεσμάτων [15%]

Το TrASH θα πρέπει να προβάλει τα post που ανακτήθηκαν για ένα ερώτημα κατά σειρά σχετικότητας (από το πιο σχετικό προς το λιγότερο σχετικό) μαζί με το σκορ σχετικότητας, και ένα μικρό απόσπασμα από το post που περιέχει τις λέξεις κλειδιά που υπέβαλλε ο χρήστης (κατά τη συνήθη πρακτική των μηχανών αναζήτησης). Αν ο χρήστης επιλέξει ένα συγκεκριμένο post από τη λίστα των αποτελεσμάτων θα πρέπει να προβάλλεται το εν λόγω post και το σύνολο των πεδίων του με εύληπτο και περιεκτικό τρόπο. Η υλοποίησή σας θα πρέπει να δίνει στο χρήστη τη δυνατότητα να περιορίσει τον αριθμό των αποτελεσμάτων που θα λάβει για κάθε ερώτημα στα K πρώτα/πιο σχετικά (να εμφανίζει δηλαδή μόνο τα top-K αποτελέσματα ανάλογα με το K που έδωσε ή έχει θέσει ως προ-ρύθμιση ο χρήστης).

Σημειώστε ότι το ElasticSearch, για κάθε ερώτημα που υποβάλλεται υπολογίζει και επιστρέφει αυτόματα το σκορ με τις αποθηκευμένες εγγραφές, επομένως η δική σας δουλειά είναι κυρίως στην σωστή παρουσίαση των αποτελεσμάτων και όχι στον υπολογισμό των σκορ. Αν και στη βασική του υλοποίηση το ElastiSearch δεν υπολογίζει σκορ ομοιότητας συνημιτόνου αλλά το σκορ για το μοντέλο BM25 (το οποίο έχουμε συζητήσει ακροθιγώς στην τάξη), για τους σκοπούς της παρούσας άσκησης αυτό είναι αρκετό καθώς μας επιτρέπει να ταξινομήσουμε τα αποτελέσματα που επιστρέφονται στο χρήστη. Επομένως, στο κομμάτι αυτό η δική σας δουλειά είναι κυρίως στην σωστή παρουσίαση των αποτελεσμάτων και όχι στον υπολογισμό των σκορ.

Τέλος, φροντίστε το γραφικό περιβάλλον που θα σχεδιάσετε να είναι απλό και κατανοητό, να δίνει στο χρήστη τη δυνατότητα να υποβάλλει με έναν εύκολο τρόπο το (πιθανόν σύνθετο) ερώτημά του, ενώ θα πρέπει να υποστηρίζει κι έναν εύληπτο τρόπο παρουσίασης των αποτελεσμάτων.

Γραπτή αναφορά [15%]

Η αναφορά σας θα πρέπει να έχει έκταση τουλάχιστον 5 σελίδες (χωρίς το εξώφυλλο και τις αναφορές/παραπομπές) και θα πρέπει να περιέχει λεπτομέρειες της υλοποίησής σας, οδηγίες εκτέλεσης του κώδικά σας, παραδείγματα επίδειξης του προγράμματός σας, επεξήγηση των βασικών στοιχείων του προγράμματός σας (user manual), καθώς και μία ανάλυση με ενδιαφέροντα στοιχεία/διαπιστώσεις από αυτό το (ομολογουμένως ασυνήθιστο) σύνολο δεδομένων. Μην συμπεριλάβετε στην αναφορά σας μέρος ή σύνολο του κώδικα, εκτός αν αφορά σε κάποιο μικρό μέρος που χρήζει (σύντομης) επεξήγησης!

Θέματα υλοποίησης και bonus

Μπορείτε στην υλοποίησή σας να χρησιμοποιήσετε κώδικα τρίτων ή έτοιμες βιβλιοθήκες (και εκτός αυτών που παρέχονται από το ElasticSearch) αρκεί να περιλαμβάνετε κατάλληλη παραπομπή της πηγής σας στην αναφορά.

Σε συνεννόηση με το διδάσκοντα μπορείτε να πάρετε μέχρι 20% bonus για επιπλέον χαρακτηριστικά ή λειτουργικότητα που θα υλοποιήσετε. Τέτοια μπορεί να είναι:

- **[+20%]** να προστίθενται εγγραφές αυτόματα κάνοντας scraping την πληροφορία από site μου μαζεύουν πληροφορία από διάφορα κοινωνικά δίκτυα όπως το RollCall, ή τα ίδια τα κοινωνικά δίκτυα του Donald Trump όπως το Facebook (σχετικά δύσκολη και τεχνική η συλλογή), το X (πρώην Twitter με API για περιορισμένο αριθμό tweets) ή το Truth (την πραγματική φωνή της Αμερικής σύμφωνα με τον ίδιο),
- **[+10%]** η υλοποίηση/ενσωμάτωση κώδικα για υπολογισμό του σκορ με βάρυνση tf-idf και ομοιότητα συνημιτόνου,
- **[+10%]** η υλοποίηση του προγράμματος ως web εφαρμογή,
- **[+10%]** η δυναμική προσθήκη/διαγραφή πεδίων, ώστε το TrASH να μπορεί να διαχειριστεί και άλλου είδους πληροφορία με διαφορετικά πεδία όπως (ενδεικτικά) tweets, λόγους ή σχετική ειδησεογραφία
- ή μία δική σας πρόταση βελτίωσης για επιπλέον χαρακτηριστικά ή λειτουργικότητα που θα υλοποιήσετε.

Καλή δουλειά!