APACHE LUCENE

Μεντόνκα Ιζαμπέλ – 3100109
Πίτσιος Σταμάτης – 3100153

Team-IS

Τι είναι η LUCENE (επανάληψη)

- Η LUCENE είναι μια υψηλής απόδοσης μηχανή αναζήτησης ***κειμένου*** .
- Γλώσσα υλοποίησης : JAVA
- Δεν είναι μια πλήρης εφαρμογή αλλά χρησιμοποιείται ως ***βιβλιοθήκη*** ή ***API*** .

Παράλειψη στη Lucene (1)

- Η ευρετηρίαση και η αναζήτηση μπορούν να γίνουν σε διαφορετικές χρονικές στιγμές.
- Κατά την ευρετηρίαση ορίζουμε τον αναλυτή που επιθυμούμε για την επεξεργασία των κειμένων καθώς και προαιρετικά την συνάρτηση ομοιότητας που θέλουμε.
- Όμοιες ενέργειες κατά την αναζήτηση.

Παράλειψη στη Lucene (2)

- Ωστόσο τη στιγμή της αναζήτησης δεν γίνεται έλεγχος αν ο αναλυτής και η συνάρτηση ομοιότητας που έχουν οριστεί, ταιριάζουν με αυτούς που έχουν οριστεί κατά την ευρετηρίαση.
- Αν είναι διαφορετικοί, η Lucene θα συνεχίσει τη διαδικασία κανονικά.
- **Τι συνέπειες όμως μπορεί να έχει αυτό;;;;;**

Σημαντικές έννοιες
Meet our Metrics

- **Precision (Ακρίβεια)** : Ο αριθμός των σχετικών ανακτηθέντων κειμένων προς τον αριθμό των συνολικών ανακτηθέντων.
- **MAP (Mean Average Precision)** : Η μέση ακρίβεια που επιτυγχάνουμε λαμβάνοντας υπόψη όλα τα ερωτήματα που έχουν γίνει.
- **P@10** : Η ακρίβεια στα πρώτα 10 ανακτηθέντα κείμενα.
- **P@20** : Η ακρίβεια στα πρώτα 20 ανακτηθέντα κείμενα.

Ένα παράδειγμα

Χρήση της συλλογής MEDLARS (1033 κείμενα και 30 ερωτήματα)

Κατά την ευρετηρίαση **και** την αναζήτηση έγινε χρήση του αναλυτή *EnglishAnalyzer* και της συνάρτηση ομοιότητας *DefaultSimilarity*.

- Αποτελέσματα :

MAP	P@10	P@20	Σχετικά ανακτηθέντα κείμενα
49,93%	62,33%	53,33%	521/696

Ένα παράδειγμα

Τώρα, για το ίδιο ευρετήριο, κατά την αναζήτηση έγινε χρήση του αναλυτή *StandardAnalyzer* και της συνάρτηση ομοιότητας BM25 similarity.

- Αποτελέσματα :

MAP	P@10	P@20	Σχετικά ανακτηθέντα κείμενα
49,93%	62,33%	53,33%	521/696
36,8%	**51%**	**41%**	**427/696**

Μεγάλη διαφορά!!

Συμπεράσματα

Χαμηλά αποτελέσματα στην αξιολόγηση μπορεί να σημαίνουν τα εξής :

- Προβληματική συλλογή, οπότε χρειάζεται επέμβαση σε αυτήν,
- Διαφορετική επιλογή αναλυτή ή/και συνάρτησης ομοιότητας

Στη δεύτερη περίπτωση, ο χρήστης μπορεί να μην είναι σε θέση να ξέρει ή να μην θυμάται τις παραπάνω πληροφορίες, εφόσον η Lucene δεν παρέχει πρόσβαση σε αυτές.

Συμπεράσματα

Χαμηλά αποτελέσματα στην αξιολόγηση μπορεί να σημαίνουν τα εξής

- Προβληματική συλλογή, οπότε χρειάζεται επέμβαση σε αυτήν,
- Διαφορετική επιλογή αναλυτή ή/και συνάρτησης ομοιότητας

Στη δεύτερη περίπτωση ο χρήστης μπορεί να μην είναι σε θέση να ξέρει ή να μην θυμάται τις παραπάνω πληροφορίες, εφόσον η Lucene δεν παρέχει πρόσβαση σε αυτές.

Εδώ θα παρέμβουμε εμείς!!

Σκοπός της συνεισφοράς

- Σκοπός μας είναι να δημιουργούμε κατά την ευρετηρίαση ένα xml αρχείο όπου θα αποθηκεύουμε πληροφορίες(metadata) που αφορούν:
  - τη συνάρτηση ομοιότητας
  - τον αναλύτη
  - την έκδοση της Lucene
- Κατά την αναζήτηση θα ανακτούμε τις πληροφορίες αυτές και θα τις συγκρίνουμε με αυτές που έχουν οριστεί για την αναζήτηση.
- Αν αυτές δεν ταιριάζουν, θα εμφανίζεται ένα προειδοποιητικό μήνυμα στο χρήστη.

Παράδειγμα xml αρχείου

```
<?xml version="1.0" encoding="UTF-8" ?>

<indexInfo>

    <uses-version>
        LUCENE_47
    </uses-version>

    <uses-analyzer>
        EnglishAnalyzer
    </uses-analyzer>

    <uses-similarity>
        DefaultSimilarity
    </uses-similarity>

</indexInfo>
```

Η συνεισφορά (1)

- Δημιουργία νέας κλάσης ***MetaDataWriter***, υπεύθυνη για τη δημιουργία του xml αρχείου.

Θα περιέχει :

- ***private IndexWriterConfig iwc*** : αντικείμενο που κρατάει τις πληροφορίες που αφορούν την ευρετηρίαση.
- ***public void writeMetaData()*** : μέθοδος υπεύθυνη για την εγγραφή του xml αρχείου.

Η συνεισφορά (2)

- Δημιουργία νέας κλάσης ***MetaDataReader***, υπεύθυνη για να διαβάζει το xml αρχείο.

Θα περιέχει :

- ***private Analyzer analyzer***
- ***private Similarity similarity***
- ***private Version version***
- ***private MetaDataParser parser***
- ***public void readMetaData()*** : μέθοδος υπεύθυνη για την ανάγνωση του xml αρχείου.
- **public boolean usesSameAnalyzer()**: μέθοδος υπεύθυνη για τη σύγκριση των αναλυτών.
- **public boolean usesSameSimilarity()**: μέθοδος υπεύθυνη για τη σύγκριση των συναρτήσεων ομοιότητας.
- **public boolean usesSameVersion()**: μέθοδος υπεύθυνη για τη σύγκριση των εκδόσεων της Lucene.

Η συνεισφορά (3)

- Δημιουργία νέας κλάσης ***MetaDataParser***, που θα αρχικοποιεί τη διαδικασία του parsing του xml αρχείου χρησιμοποιώντας το API SAX(Simple API for XML).

Θα περιέχει :

- ***public Object[] parse()*** : μέθοδος υπεύθυνη για την αρχικοποίηση του parsing του xml αρχείου. Θα επιστρέφει ένα πίνακα αντικειμένων Object όπου στην πρώτη θέση θα βρίσκεται η έκδοση της Lucene, στη δεύτερη θέση ο αναλυτής και στην τρίτη η συνάρτηση ομοιότητας.

Η συνεισφορά (4)

- Δημιουργία νέας κλάσης ***MetaDataHandler***, που θα κληρονομεί τη ***DefaultHandler*** και θα είναι υπεύθυνη για τον χειρισμό του xml αρχείου.

Θα περιέχει τις υλοποιήσεις των παρακάτω μεθόδων που κληρονομούνται από τη ***DefaultHandler***:

- ***public void startElement***
- ***public void characters***
- ***public void endElement***

Η συνεισφορά (5)

- Γενικά για τη δημιουργία των ευρετηρίων πρέπει να αρχικοποιούμε ένα αντικείμενο της κλάσης **IndexWriter**. Εμείς θα επέμβουμε στον κατασκευαστή της κλάσης αυτής, προσθέτωντας την αρχικοποιήση του αντικειμένου **MetaDataWriter** και καλώντας τη μέθοδο **writeMetaData()**.

- Επειδή δεν υπάρχει κάποια get μέθοδος για την έκδοση της Lucene, θα προσθέσουμε τη μέθοδο **getVersion()** στην κλάση **LiveIndexWriterConfig**.

Η συνεισφορά (6)

- Για τη σύγκριση μεταξύ των συναρτήσεων ομοιότητας που χρησιμοποιούνται, θα αξιοποιούμε τις μεθόδους της κλάσης **MetaDataReader** στις διάφορες μεθόδους **search()** που έχει η κλάση **IndexSearcher**.

- Για τη σύγκριση μεταξύ των εκδόσεων της Lucene και των αναλυτών, θα προβούμε σε παρόμοιες ενέργειες με παραπάνω στη μέθοδο **init ()** της κλάσης **QueryParserBase**.

Τεχνικές συνιστώσες του έργου

- Όλες οι αλλάγες που θα πραγματοποιηθούν θα είναι σε **java**, εφόσον και το έργο είναι εξ'ολοκλήρου γραμμένο σε java.
- Σύστημα δόμησης του έργου : **Ant**
- Σύστημα διαχείρισης εκδόσεων : **Github**

Διαχείριση Δουλειάς

- Προβλεπόμενος Χρονοπρογραμματισμός :

Κώδικας	Εβδομάδα 7/4-13/4 Εβδομάδα 14/4-20/4 Εβδομάδα 21/4-27/4
Υλοποίηση Ελέγχων	Εβδομάδα 28/4-4/5 Εβδομάδα 5/4-11/5

- Κατανομή δουλειάς :
  - MetaDataWriter, MetaDataHandler : Σταμάτης
  - MetadataReader, MetaDataParser : Ιζαμπέλ
- Συνεργασία ομάδας : email, github, teamviewer

Thank you!!