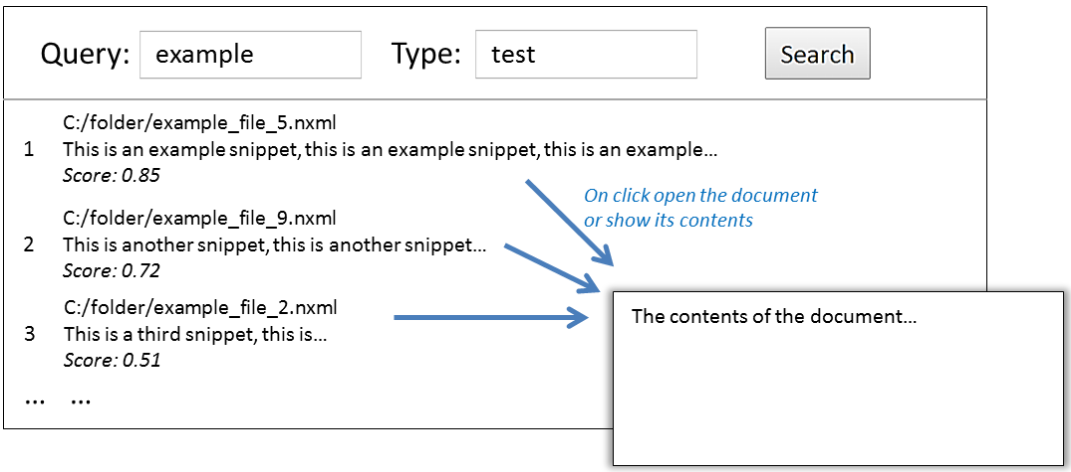

Εικόνα 2: *Παράδειγμα γραφικής διεπαφής*

ΦΑΣΗ Β (40 μονάδες)

Στη συγκεκριμένη φάση καλείστε να αξιολογήσετε την **αποτελεσματικότητα** του συστήματός σας.

Για κάθε ιατρικό θέμα στο αρχείο **topics.xml**, σας δίνεται:

- ένα σύνολο εγγράφων της συλλογής που είναι πολύ σχετικά για την απάντηση του αντίστοιχου θέματος
- ένα σύνολο εγγράφων της συλλογής που είναι σχετικά για την απάντηση του αντίστοιχου θέματος
- ένα σύνολο εγγράφων της συλλογής που δεν είναι σχετικά για την απάντηση του αντίστοιχου θέματος

Οι παραπάνω πληροφορίες δίνονται στο TSV (Tab-Separated Values) αρχείο **qrels.txt**. Κάθε γραμμή αυτού του αρχείου περιέχει 4 στοιχεία:

1. *topic number:* αριθμός από 1 έως 30 που αναπαριστά το αντίστοιχο ιατρικό θέμα του αρχείου **topics.xml**)
2. *αριθμός 0* (δεν χρησιμοποιείται)
3. *document PMCID*: αναγνωριστικό PMC βιοϊατρικού άρθρου από τη συλλογή **Medical Collection**
4. *relevance score*: η σχετικότητα του βιοϊατρικού άρθρου για την απάντηση του ιατρικού θέματος (0 = μη σχετικό, 1 = σχετικό, 2 = πολύ σχετικό)

Για παράδειγμα, η γραμμή «**1 0 1033658 0**» σημαίνει ότι το έγγραφο της συλλογής με PMCID "1033658 " δεν είναι σχετικό για το ιατρικό θέμα με αριθμό 1.

Για πολλά από τα έγγραφα τις συλλογής δεν γνωρίζουμε αν είναι σχετικά ή όχι για την απάντηση ενός ή περισσοτέρων θεμάτων. Γι' αυτό τον λόγο πρέπει να χρησιμοποιήσετε κάποιες μετρικές αξιολόγησης ιδανικά σχεδιασμένες για τέτοιες περιπτώσεις. Οι μετρικές που πρέπει να χρησιμοποιήσετε είναι οι παρακάτω:

- bpref [1]
- AveP' [2]
- NDCG' [2]

Για να καταλάβετε και να υλοποιήσετε αυτές τις μετρικές πρέπει να διαβάσετε τα παρακάτω 2 άρθρα:

- [1] Chris Buckley and Ellen M. Voorhees, *"Retrieval evaluation with incomplete information.",* Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2004.
- [2] Tetsuya Sakai, *"Alternatives to bpref.",* Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2007.

Χρησιμοποιώντας τις παραπάνω μετρικές, θα αξιολογήσετε την αποτελεσματικότητα του συστήματός σας.
Δημιουργήστε ένα πρόγραμμα που θα αυτοματοποιεί τον υπολογισμό των παραπάνω μέτρων. Αρχικά, το πρόγραμμά σας πρέπει να διαβάζει α) το αρχείο με τα αποτελέσματα σας **(results.txt)**, και β) το αρχείο με τα μερικά αποτελέσματα

συνάφειας (**qrels.txt**). Έπειτα θα υπολογίζει τις τιμές των παραπάνω μετρικών **για κάθε ιατρικό θέμα**, και θα τις αποθηκεύει σε ένα TSV αρχείο με όνομα «**eval_results.txt**»στην μορφή:

`TOPIC_NO BPREF_VALUE AVEP_VALUE NDCG_VALUE`

Δείτε τα αποτελέσματα και κάντε ό,τι παραλλαγή νομίζετε στο σύστημά σας (π.χ. στη βάρυνση του ευρετηρίου, στον επεξεργαστή επερωτήσεων, στη συνάρτηση υπολογισμού του βαθμού συνάφειας, κλπ.) **ώστε να μεγιστοποιήσετε την αποτελεσματικότητα του συστήματός σας και να αυξήσετε την πιθανότητα να βγείτε νικητές!**

Συντάξτε σχετική αναφορά που να περιγράφει τις καλύτερες επιδόσεις του συστήματος σας και τι κάνατε για να τις επιτύχετε. Επίσης, στην αναφορά πρέπει να αναλύετε τα αποτελέσματα της πειραματικής αξιολόγησης (με ποια ιατρικά θέματα το σύστημά σας τα πήγε καλά, με ποια όχι, που μπορεί να οφείλετε η επιτυχία/αποτυχία, κτλ.). Επιπλέον θα πρέπει να δοθούν σχετικά στατιστικά στοιχεία, π.χ. median/average values, min, max, κ.ο.κ. Για την κατασκευή των γραφημάτων μπορείτε να χρησιμοποιήσετε τις δυνατότητες του excel.

Το καλύτερο σύστημα θα είναι αυτό με τις περισσότερες «νίκες» σε ένα **νέο** σύνολο ιατρικών θεμάτων (για ένα ιατρικό θέμα, το σύστημα που κερδίζει είναι αυτό με το **υψηλότερο άθροισμα** των τιμών των τριών μετρικών, ενώ σε περίπτωση ισοπαλίας, θα υπολογίζονται οι δεύτερες θέσεις, κ.ο.κ.)

➔ **ΠΑΡΑΔΟΤΕΑ**: Ένα αρχείο <<AM1-AM2>>.zip το οποίο να περιέχει

- /doc/report.{doc|pdf}: Η γραπτή αναφορά στην οποία πρέπει να περιγράψατε τι ακριβώς κάνατε και τα αποτελέσματα της πειραματικής αξιολόγησης (χρησιμοποιείστε το πρότυπο που σας έχει δοθεί - HY463_Report_Template_2017).
- /doc/**eval_results.txt**: Το αρχείο **eval_results.txt** όπως παράγεται από το πρόγραμμά σας.
- /src: Με τον κώδικά σας **(ΠΡΟΣΟΧΗ: Να περιέχει τα .java αρχεία, όχι τα .class, ή ιδανικά ολόκληρο το Netbeans ή Eclipse project)**
- /dist: Με τα αρχεία .jar τα οποίο πρέπει να επαρκούν για την εκτέλεση του προγράμματός σας. Είναι σημαντικό για την τελική βαθμολόγηση της εργασίας σας η ομαλή και εύκολη εκτέλεσή της.

Καλή εργασία!

ΠΑΡΑΡΤΗΜΑΤΑ

ΠΑΡΑΡΤΗΜΑ Α – Η ΣΥΛΛΟΓΗ

Η συλλογή που θα χρησιμοποιήσετε περιέχει άρθρα σχετικά με την βιοϊατρική τα οποία έχουν εξαχθεί από τη ψηφιακή βάση δεδομένων «PubMed Central» (PMC). Κάθε άρθρο της συλλογής αναπαριστάται ως ένα NXML αρχείο (XML αρχείο κωδικοποιημένο με χρήση της βιβλιοθήκης «NLM Journal Archiving and Interchange Tag Library» και αναγνωρίζεται μοναδικά από τον αριθμό PMCID (το όνομα κάθε αρχείου είναι στην ουσία ο αριθμός PMCID του αντίστοιχου άρθρου).

Κάθε NXML αρχείο περιλαμβάνει πολλές ετικέτες όπως: αναγνωριστικά του άρθρου, αναγνωριστικό περιοδικού, τίτλος περιοδικού, εκδότης περιοδικού, τίτλος άρθρου, περίληψη άρθρου, συγγραφείς, κυρίως σώμα άρθρου, ημερομηνία δημοσίευσης, αναφορές, και πολλά άλλα.

Στα πλαίσια αυτής της εργασίας μας ενδιαφέρουν οι παρακάτω 8 ετικέτες:

- *Αναγνωριστικό PMCID*
- *Τίτλος άρθρου*
- *Συγγραφείς του άρθρου*
- *Περίληψη άρθρου*
- *Κυρίως σώμα άρθρου*

- *Κατηγορίες άρθρου*
- *Περιοδικό που δημοσιεύτηκε*
- *Εκδότης περιοδικού*

Στο Παράρτημα Β θα βρείτε κώδικα για την ανάγνωση των παραπάνω ετικετών. Φυσικά, αν το επιθυμείτε, είστε ελεύθεροι να χρησιμοποιήσετε όποιες άλλες ετικέτες θέλετε.

ΠΑΡΑΡΤΗΜΑ Β – ΑΝΑΓΝΩΣΗ ΒΙΟΪΑΤΡΙΚΩΝ ΑΡΘΡΩΝ | ΔΙΑΧΩΡΙΣΜΟΣ ΑΛΦΑΡΙΘΜΗΤΙΚΟΥ ΣΕ ΛΕΞΕΙΣ

Για την ανάγνωση ενός βιοϊατρικού άρθρου μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη «**BioReader.jar**». Αφού προσθέσετε την βιβλιοθήκη στο πρόγραμμά σας, με τον παρακάτω κώδικα μπορείτε να διαβάσετε τις ετικέτες ενός άρθρου:

```
import gr.uoc.csd.hy463.NXMLFileReader;
import java.io.File;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashSet;

public class MYEXAMPLE {

    public static void main(String[] args) throws UnsupportedEncodingException, IOException {

        File example = new File("C:\\dataset\\clinic\\3536594.nxml");

        NXMLFileReader xmlFile = new NXMLFileReader(example);
        String pmcid = xmlFile.getPMCID();
        String title = xmlFile.getTitle();
        String abstr = xmlFile.getAbstr();
        String body = xmlFile.getBody();
        String journal = xmlFile.getJournal();
        String publisher = xmlFile.getPublisher();
        ArrayList<String> authors = xmlFile.getAuthors();
        HashSet<String> categories =xmlFile.getCategories();

        System.out.println("- PMC ID: " + pmcid);
        System.out.println("- Title: " + title);
        System.out.println("- Abstract: " + abstr);
        System.out.println("- Body: " + body);
        System.out.println("- Journal: " + journal);
        System.out.println("- Publisher: " + publisher);
        System.out.println("- Authors: " + authors);
        System.out.println("- Categories: " + categories);

    }
}
```

Το παρακάτω κομμάτι κώδικα εκτυπώνει όλες τις λέξεις ενός αλφαριθμητικού:

```
String delimiter = "\t\n\r\f ";

String line = "hello, my name is Pavlos, how are you?";
StringTokenizer tokenizer = new StringTokenizer(line, delimiter);
while(tokenizer.hasMoreTokens() ) {
       String currentToken = tokenizer.nextToken();
       System.out.println(currentToken);
}
```

ΠΑΡΑΡΤΗΜΑ Γ – ΠΡΟΣΒΑΣΗ ΣΤΑ ΑΡΧΕΙΑ ΕΝΟΣ ΦΑΚΕΛΟΥ

Ενδεικτική πρόσβαση σε όλα τα αρχεία ενός φακέλου (συμπεριλαμβανομένων των αρχείων σε υποφακέλους αναδρομικά):

```
import java.io.File;

public class ReadAllFiles {

    public static void main(String[] args) {
        File folder = new File("C:\\dataset\\ clinic\\");
        listFilesForFolder(folder);

    }

    public static void listFilesForFolder(File folder) {
        for (File fileEntry : folder.listFiles()) {
            if (fileEntry.isDirectory()) {
                listFilesForFolder(fileEntry);
            } else {
                System.out.println(fileEntry.getAbsolutePath());
            }
        }
    }
}
```

ΠΑΡΑΡΤΗΜΑ Δ – STEMMING

Ενδεικτική χρήση του `Stemmer` της μηχανής αναζήτησης mitos.

```
import mitos.stemmer.Stemmer;

public class TestStemmer {

       public static void main(String[] args){
              Stemmer.Initialize();
              System.out.println(Stemmer.Stem("ending"));
              System.out.println(Stemmer.Stem("συγχωνευμένος"));

       }
}
```

ΠΑΡΑΡΤΗΜΑ Ε – RANDOM ACCESS FILE

Ενδεικτική ανάγνωση και εγγραφή σε random access αρχείο:

```
import java.io.*;

public class  WRFile
{
    public static void main(String[] args)
    {
         RandomAccessFile file = null;
         try {
             file = new RandomAccessFile("rand.txt","rw");

             //Writing to the file
             file.writeChar('A');
             file.writeChar('B');
             file.writeChar('C');
```

```
            file.writeChar('D');

            file.seek(0);     // get first item
            System.out.println(file.readChar());

            file.seek(4);  //get third item (char size  = 2 byte, 2*2)
            System.out.println(file.readChar());


            file.close();
           } catch(Exception e) {}
       }
}
```

ΠΑΡΑΡΤΗΜΑ ΣΤ – ΑΝΑΓΝΩΣΗ ΙΑΤΡΙΚΩΝ ΑΝΑΦΟΡΩΝ ΑΠΟ ΑΡΧΕΙΟ

Για την ανάγνωση ενός του αρχείου με τις ιατρικές αναφορές μπορείτε να χρησιμοποιήσετε την βιβλιοθήκη «**BioReader.jar**». Αφού προσθέσετε την βιβλιοθήκη στο πρόγραμμά σας, με τον παρακάτω κώδικα μπορείτε να διαβάσετε τις όλες τις ιατρικές αναφορές:

```
import gr.uoc.csd.hy463.Topic;
import gr.uoc.csd.hy463.TopicsReader;
import java.util.ArrayList;

public class MYEXAMPLE {

    public static void main(String[] args) throws Exception {

        ArrayList<Topic> topics = TopicsReader.readTopics("C:\\dataset\\ topics.xml");
        for (Topic topic : topics) {
            System.out.println(topic.getNumber());
            System.out.println(topic.getType());
            System.out.println(topic.getSummary());
            System.out.println(topic.getDescription());
            System.out.println("---------");
        }

    }
}
```