



APACHE LUCENE

Μεντόνκα Ιζαμπέλ – 3100109
Πίτσιος Σταμάτης – 3100153

Team-IS

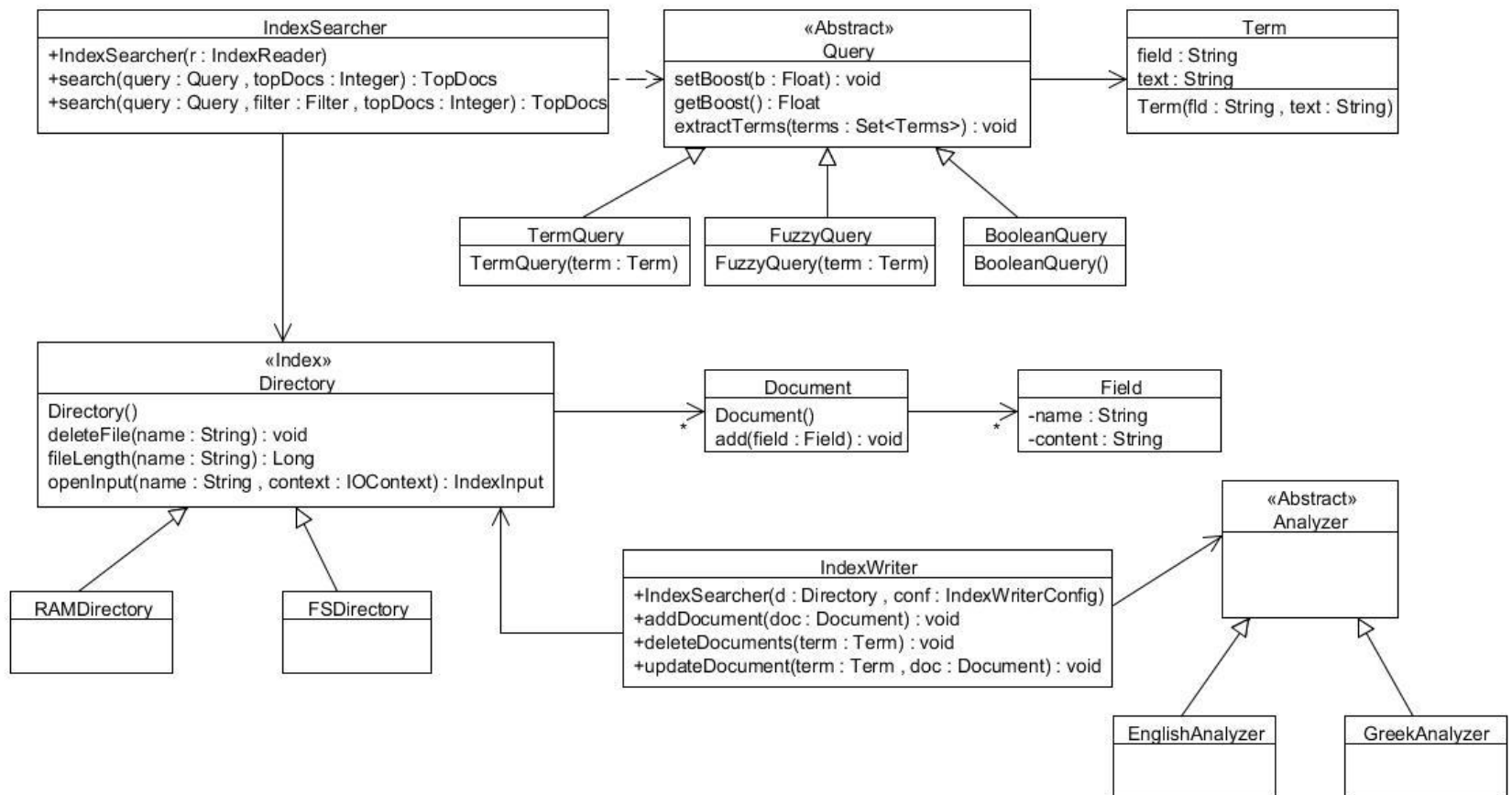
Τι είναι η LUCENE (επανάληψη)

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

Κατανόηση έργου

- Το αρχιτεκτονικό πρότυπο που ακολουθεί είναι Component - based.
- Έχει ξεχωριστά πακέτα που διαχωρίζουν τον κώδικα κάθε λειτουργίας της Lucene.
- Η κατανόηση του έργου ήταν σχετικά εύκολη, καθώς είχαμε ξαναχρησιμοποιήσει το εργαλείο αυτό.

Διάγραμμα Κλάσεων



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.
- **Αυτό συνεπάγεται ότι η αναζήτηση δεν θα έχει τα βέλτιστα αποτελέσματα.**

Στόχος της συνεισφοράς

- Στόχος μας ήταν να δημιουργήσουμε κατά την ευρετηρίαση ένα xml αρχείο όπου θα αποθηκεύαμε πληροφορίες(metadata) που αφορούν:

 - τη συνάρτηση ομοιότητας
 - τον αναλύτη
 - την έκδοση της Lucene
- Κατά την αναζήτηση ανακτούμε τις πληροφορίες αυτές και τις συγκρίνουμε με αυτές που έχουν οριστεί για την αναζήτηση.
- Αν αυτές δεν ταιριάζουν, εμφανίζεται ένα προειδοποιητικό μήνυμα στο χρήστη.

Ένα παράδειγμα

Χρήση της συλλογής MEDLARS (1033 κείμενα και 30 ερωτήματα)

	Ευρετηρίαση	Αναζήτηση	Αποτελέσματα
Εκτέλεση 1	<ul style="list-style-type: none"> EnglishAnalyzer Default Similarilty Version 4.8 	<ul style="list-style-type: none"> EnglishAnalyzer Default Similarilty Version 4.8 	521/696
Εκτέλεση 2	<ul style="list-style-type: none"> EnglishAnalyzer Default Similarilty Version 4.8 	<ul style="list-style-type: none"> StopAnalyzer BM25 Version 5.0 	427/696

Ανάλυση εκτέλεσης 2 (Προ συνεισφοράς)

- Κατά τη δεύτερη εκτέλεση , πριν κάνουμε τη συνεισφορά μας , αυτό που θα φαινόταν στην έξοδο εκτέλεσης του προγράμματος είναι το εξής :

```
Indexing to directory 'index_bigrams'. Please wait...
Time took to create index : 2.833 seconds.
```

```
Starting to query Lucene Engine...
```

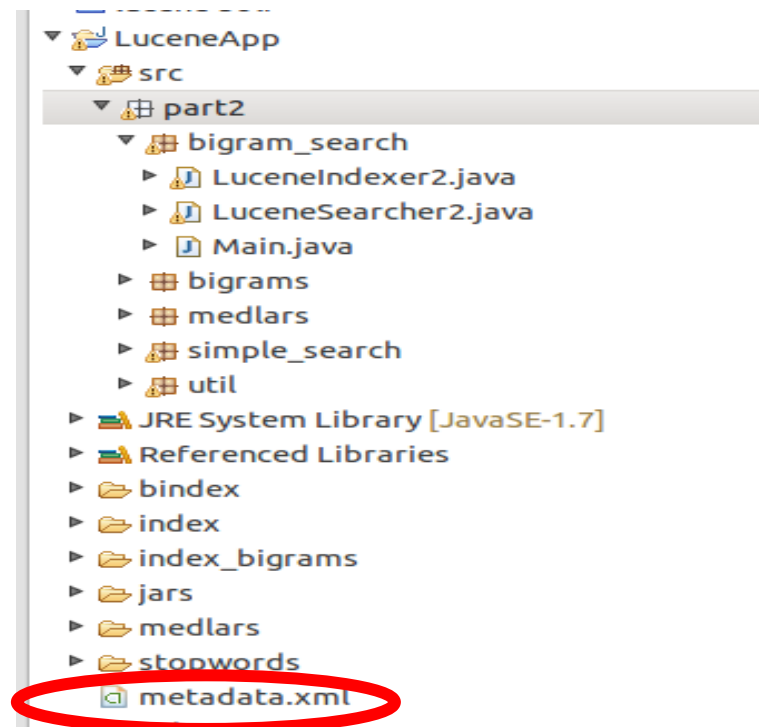
```
Time took to get the top 100 answers for all the queries is : 0.46 seconds.
```

```
|
```

- Δηλαδή δεν θα καταλαβαίναμε γιατί κατά την αναζήτηση θα είχαμε πιο «φτωχά» αποτελέσματα.

Ανάλυση εκτέλεσης 2 (Μετά συνεισφοράς)

- Αρχικά κατά την αναζήτηση δημιουργείται το xml αρχείο με τα metadata.



Ανάλυση εκτέλεσης 2 (Μετά συνεισφοράς)

- Περιεχόμενα xml αρχείου

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<indexinfo>  
  <uses-version>  
    LUCENE_48  
  </uses-version>  
  
  <uses-analyzer>  
    org.apache.lucene.analysis.en.EnglishAnalyzer  
  </uses-analyzer>  
  
  <uses-similarity>  
    DefaultSimilarity  
  </uses-similarity>  
  
</indexinfo>
```

Ανάλυση εκτέλεσης 2 (Μετά συνεισφοράς)

- Αποτελέσματα της εκτέλεσης στην έξοδο του προγράμματος ,μετά τη συνεισφορά μας , είναι τα εξής :

```
Indexing to directory 'index bigrams'. Please wait...
Time took to create index : 3.229 seconds.
```

```
Starting to query Lucene Engine...
```

```
WARNING: The version used for searching is not the same with the one used for indexing.
Expected LUCENE_48 but LUCENE_50 is being used.
```

```
WARNING: The analyzer used for searching is not the same with the one used for indexing.
Expected org.apache.lucene.analysis.en.EnglishAnalyzer but org.apache.lucene.analysis.core.StopAnalyzer is being used.
```

```
WARNING: The similarity function used for searching is not the same with the one used for indexing.
Expected DefaultSimilarity but BM25(k1=1.2,b=0.75) is being used.
```

```
Time took to get the top 100 answers for all the queries is : 0.393 seconds.
```

Πλάτος αλλαγών

- Οι αλλαγές που κάναμε αφορούν τόσο στη δημιουργία νέων κλάσεων , όσο και στην παρέμβαση σε ήδη υπάρχουσες κλάσεις.
- Το σύνολο των αλλαγών μας, εκτείνεται σε περίπου 600 γραμμές κώδικα.
- Πιο συγκεκριμένα, οι αλλαγές που κάναμε φαίνονται παρακάτω.

Περιεχόμενα συνεισφοράς μας (1/2)

- Οι νέες κλάσεις που δημιουργήσαμε είναι οι εξής:
 - MetadataWriter
 - MetadataReader
 - MetadataHandler
 - MetadataParser
- Οι κλάσεις αυτές τοποθετήθηκαν σε ένα νέο πακέτο, το meta μέσα στο ήδη υπάρχον πακέτο core.
- Επίσης , δημιουργήσαμε μια διεπαφή (interface StringNormalizer) μέσα στο ίδιο πακέτο.

Π.x MetadataReader

```
152  /**
153   * Reads metadata information from the xml file.
154   */
155  public void readMetaData() {
156      String finalPath = (filePath.equals("")) ? fileName : (filePath + "/" + fileName);
157
158      this.parser = new MetaDataParser(finalPath);
159      String[] metadata = parser.parse();
160
161      this.indexVersion = metadata[0];
162      this.indexAnalyzer = metadata[1];
163      this.indexSimilarity = metadata[2];
164  }
165
```

Περιεχόμενα συνεισφοράς μας (2/2)

- Οι ήδη υπάρχουσες κλάσεις στις οποίες επέμβαμε είναι οι παρακάτω:
 - LiveIndexWriterConfig
 - IndexWriter
 - QueryParserBase
 - IndexSearcher
- Στις κλάσεις αυτές , προσθέσαμε ένα μικρό κομμάτι κώδικα ώστε να μπορέσουμε να αξιοποιήσουμε τις δυνατότητες της συνεισφοράς μας.

Π.x IndexSearcher

```
/**
 * This method checks if the similarity function
 * being used for searching matches the one used for indexing.
 */
protected void metaDataCompatibility() {
    metaReader.readMetaData();

    if(!metaReader.usesSameSimilarity()) {
        System.err.println("WARNING: The similarity function used for searching is not the same with the one used for indexing.");
        System.err.println("Expected " + metaReader.getIndexSimilarity() + " but " + metaReader.getSearchSimilarity() + " is being u

        this.warningShown = true;
    }
}
```

Ποιότητα υλοποίησης

- Ακολουθήσαμε τα πρότυπα γραφής κώδικα του έργου. Για παράδειγμα :
 - ✓ Τεκμηρίωση με χρήση javadoc τόσο της κάθε κλάσης, όσο και των ιδιωτικών και δημόσιων μεθόδων.
 - ✓ Άνοιγμα του bracket “{” στην ίδια γραμμή σε περιπτώσεις όπως δήλωση μεθόδου, if , while, for , κτλ.
 - ✓ Καμία κενή γραμμή μετά από τις παραπάνω περιπτώσεις.
 - ✓ Ένα κενό μεταξύ μεθόδων .
 - ✓ Ονοματοδοσία με χρήση CamelCase.

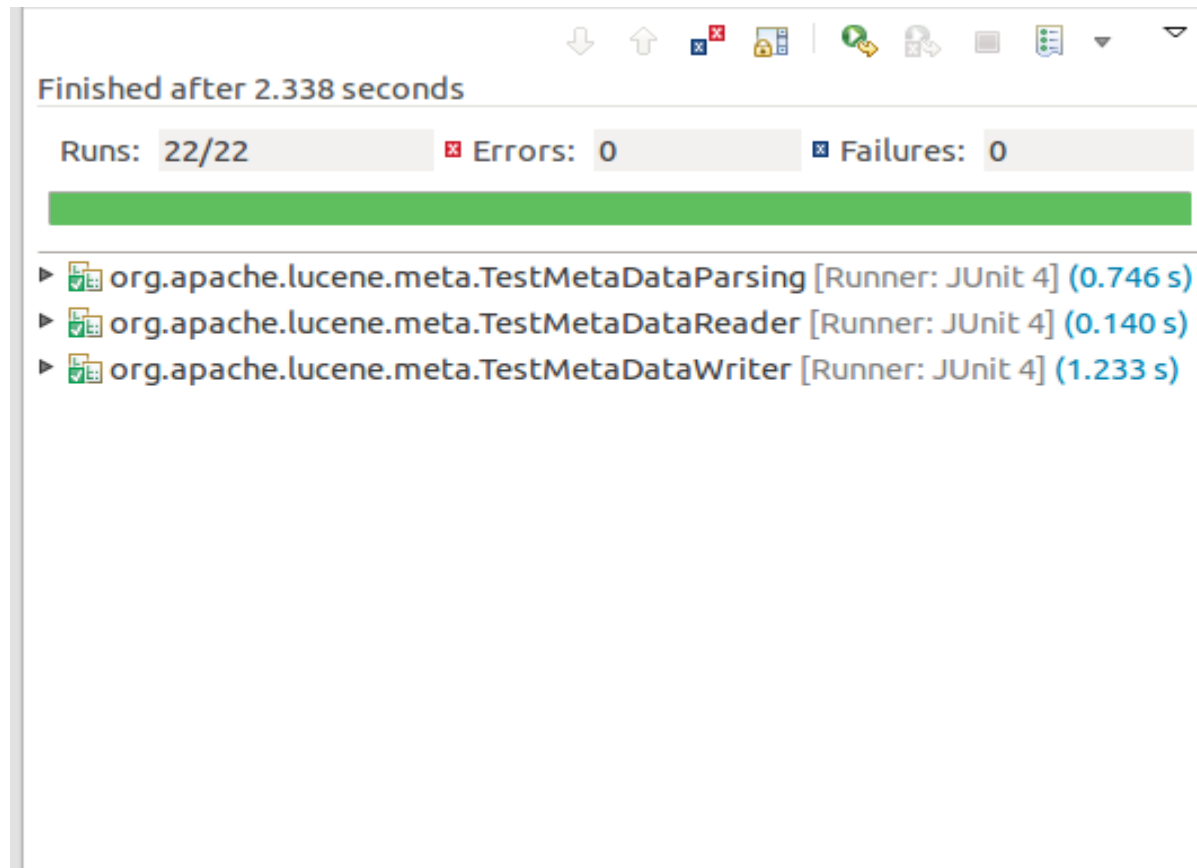
Ολοκλήρωση

- Υλοποιήσαμε όλα όσα είχαμε θέσει ως στόχους για την συνεισφορά μας.
- Επιπλέον, προσθέσαμε μικρά κομμάτια κώδικα σε ήδη υπάρχουσες κλάσεις, τα οποία δεν είχαμε προβλέψει εξ αρχής ότι θα χρειαζόμασταν. Ο κώδικας αυτός ήταν αναγκαίος προκειμένου να είναι λειτουργική και συμβατή, η συνεισφορά μας με το ήδη υπάρχον έργο.

Έλεγχος (1/3)


- Για τον έλεγχο των νέων κλάσεων και των παρεμβάσεων μας , χρησιμοποιήσαμε το εργαλείο JUnit καθώς και το εργαλείο Emma , το οποίο μας δίνει αναφορές κάλυψης κώδικα και διακλαδώσεων.
- Δημιουργήσαμε 3 κλάσεις ελέγχου που συνολικά περιέχουν 22 περιπτώσεις ελέγχου.


'Ελεγχος (2/3)




Finished after 2.338 seconds










Runs: 22/22 ✖ Errors: 0 ✖ Failures: 0

▶  org.apache.lucene.meta.TestMetaDataParsing [Runner: JUnit 4] (0.746 s)

▶  org.apache.lucene.meta.TestMetaDataReader [Runner: JUnit 4] (0.140 s)

▶  org.apache.lucene.meta.TestMetaDataWriter [Runner: JUnit 4] (1.233 s)

'Ελεγχος (3/3)

Element	Coverage	Covered Instruct	▲ Missed Instruc
▼  org.apache.lucene.meta	100.0 %	482	0
▶  MetadataHandler.java	 100.0 %	130	0
▶  MetadataParser.java	 100.0 %	35	0
▶  MetadataReader.java	 100.0 %	125	0
▶  MetadataWriter.java	 100.0 %	192	0

Οργάνωση συνεισφοράς στο GitHub (1/2)

- Το αποθετήριο που χρησιμοποιήσαμε προκειμένου να κάνουμε τις αλλαγές στο έργο είναι το **Team-IS/lucene-solr** .
- Το πρωτότυπο έργο δεν περιείχε wiki . Στο αποθετήριο μας , όμως προσθέσαμε wiki, όπου αναφέρουμε τον σκοπό των αλλαγών μας καθώς και λεπτομερή περιγραφή αυτών.

Οργάνωση συνεισφοράς στο GitHub (2/2)

- Όσον αφορά τα commits, δώσαμε ιδιαίτερη έμφαση στο περιεχόμενο τους. Προσπαθήσαμε κάθε commit να περιέχει μια μόνο συγκεκριμένη αλλαγή/προσθήκη, ώστε να έχουμε καλύτερη οργάνωση.
- Αν και συνεργαζόμασταν σε ομαδικό επίπεδο, προσπαθήσαμε τα commits να είναι όσο το δυνατόν μοιρασμένα.
- Τέλος, ως προς το branching, δεν δημιουργήσαμε κάποιο νέο κλάδο ,αλλά εργαστήκαμε στον κύριο κλάδο trunk.

Συνεργασία με την ομάδα ανάπτυξης

- Επικοινωνήσαμε με δυο άτομα από την ομάδα ανάπτυξης.
- Και οι δύο μας καλωσόρισαν και μας είπαν ότι χαρήκανε που είχαμε σκοπό να συνεισφέρουμε στο έργο.
- Όταν τους είπαμε την ιδέα της υλοποίησης μας , ο πρώτος.....



Συνεργασία με την ομάδα ανάπτυξης

- Ο δεύτερος, ήταν πιο ευγενικός, και μας συνέστησε να ανοίξουμε ένα issue στο Jira, καθώς αυτό είναι το εργαλείο που χρησιμοποιεί η κοινότητα για τη διαχείριση των αλλαγών.
- Επίσης, μας τόνισε ότι είναι καλύτερο να απευθυνθούμε γενικά στη κοινότητα της Lucene, παρά σε μεμονωμένους committers μέσω email.

Συνεργασία με την ομάδα ανάπτυξης



Lucene - Core / LUCENE-5629

1 of 1 ▲

Comparing the Version of Lucene , the Analyzer and the similarity function that are being used for indexing and searching.

Comment

Agile Board

More ▾

Reopen Issue



Export ▾

Details

Type:	+ New Feature	Status:	CLOSED
Priority:	↓ Minor	Resolution:	Not a Problem
Affects Version/s:	None	Fix Version/s:	4.8, 4.9, 5.0
Component/s:	core/index, ... (2)		
Labels:	features patch		
Environment:	Operating system : Windows 8.1 Software platform : Eclipse Kepler 4.3.2		
Lucene Fields:	New		

Description

We have observed that Lucene does not check if the same Similarity function is used during indexing and searching. The same problem exists for the Analyzer that is used. This may lead to poor or misleading results.

So we decided to create an xml file during indexing that will store information such as the Analyzer and the Similarity function that were used as well as the version of Lucene that was used. This xml file will always be available to the users.

People

Assignee:	Unassigned
Reporter:	Isabel Mendonca
Votes:	0
Watchers:	3 Stop watching this issue


Dates

Created:	24/Apr/14 13:20
Updated:	30/Apr/14 15:45
Resolved:	24/Apr/14 19:05

Time Tracking


Estimated:	<div></div> 672h
Remaining:	<div></div> 672h

Συνεργασία με την ομάδα ανάπτυξης

✓  [Ahmet Arslan](#) added a comment - 24/Apr/14 18:56


The same problem exists for the Analyzer that is used.

Can't we use different analyzers for indexing and searching? e.g. WordDelimiterFilter, SynonymFilter, NGramFilter, etc.


✓  [Erick Erickson](#) added a comment - 24/Apr/14 19:05

Closing, if you still think this is a problem we can re-open.

Allowing different analyzers at index and query time is a deliberate decision. Otherwise all the effort that went in to allowing independent index and query analysis chains could have been avoided. In particular synonyms are often defined at index time but not at query time.


✓  [Isabel Mendonca](#) added a comment - 30/Apr/14 13:35

Leaving aside the comparing part, wouldn't it be useful to store the information concerning the indexing into a separate file? In that way, the index and query analysis will still be independent and this information will be accessible to whoever wants to see it.

✓  [Ahmet Arslan](#) added a comment - 30/Apr/14 14:34

store the information concerning the indexing into a separate file?

You mean a separate file other than schema.xml?

✓  [Isabel Mendonca](#) added a comment - 30/Apr/14 15:45

What we meant is an xml file containing indexing metadata such as the similarity function and the analyzer. This xml file could be stored in a different location from where the actual index exists.



Thank you!!



A word cloud graphic featuring various terms related to search engines and technology. The words are arranged in a grid-like pattern, with some words appearing more frequently than others. The words include: ENESOLR, NETWORK, SEARCH, SYST, NNESS, IAMPSON, AVIVA, YEATS, BECH, KADO, SHAR, PLUCENE, SC, EYCA, TRINITY, GUINNES, AFTON, HES, SOLR, RUGBY, MOLL, RCH, SYST, LEIN, STER, OIF, IESON, AVIVA, BYMOLLYT, LUCENE, YE, TEMLEIN, STE, OIRE, YAYEATS, BEC, SHAV, ODER, FLEADH, SHAR, PLUCENE, SOLR, ISTEROIRE, ACHTAS, LIFFEY, CASTLES. The word 'REVOLUTION' is prominently displayed in the center, with 'REV' and 'UTION' in green and 'SOLR' and 'LUCENE' in black.