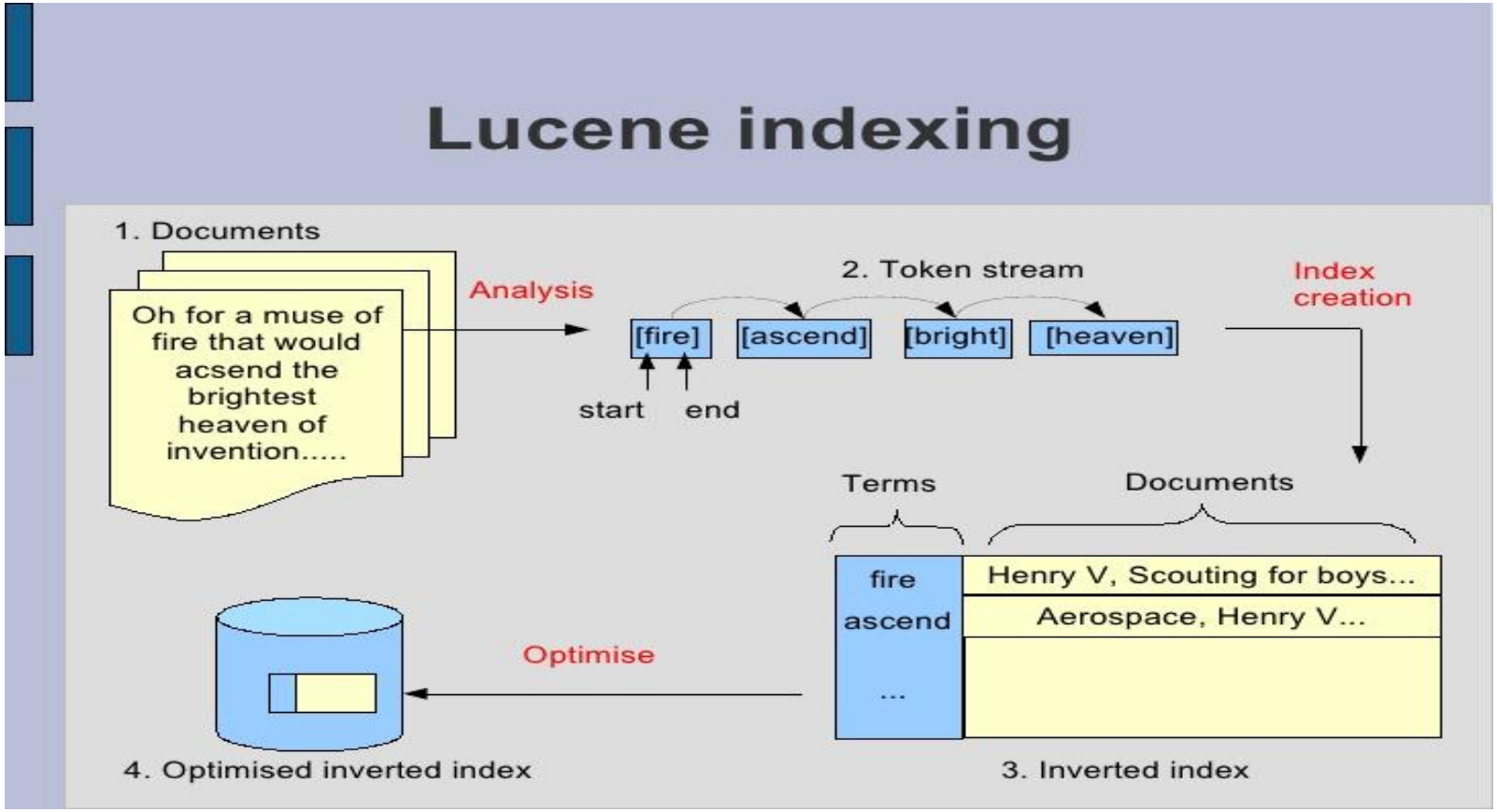
Analyse et traitement du Contenu

Indexation des Documents avec Lucene

- **Indexation – construction de l'index – classe IndexWriter**



Indexation des Documents avec Lucene

- Analyzer luceneAnalyzer = new StandardAnalyzer();
- IndexWriter indexWriter = new IndexWriter(**indexDir**, **luceneAnalyzer**);

- cette classe peut créer un nouvel index ou ouvrir un index existant et lui ajouter des documents.
- le premier paramètre : ou on stocke les fichiers d'index ;
- le deuxième paramètre : l'analyseur qui sera employé ;
- le dernier paramètre : si vraie, la classe crée un nouvel index ; si faux, il ouvre un index existant

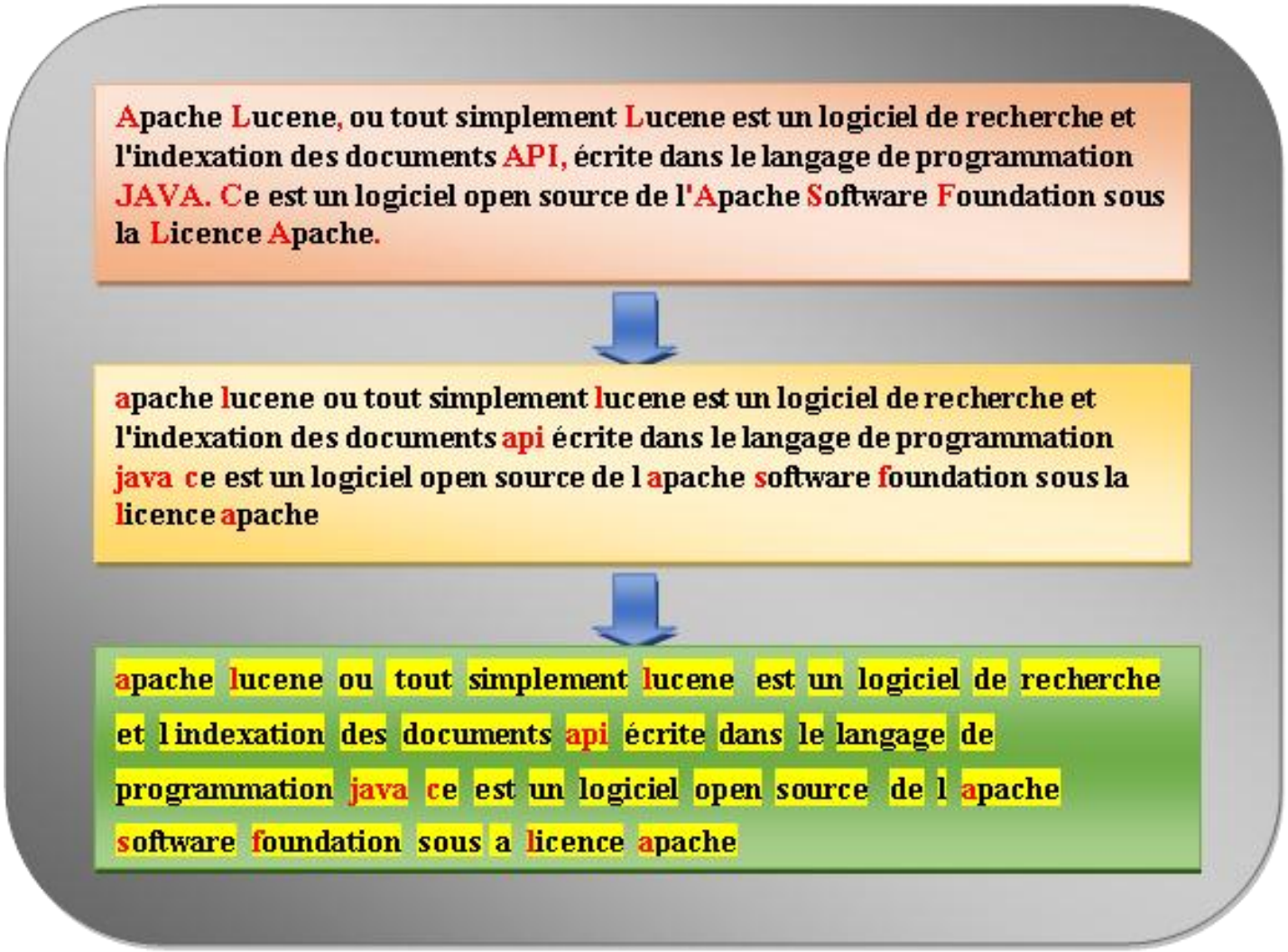
Analyse des Documents avec Lucene

- Analyzer luceneAnalyzer = new StandardAnalyzer();

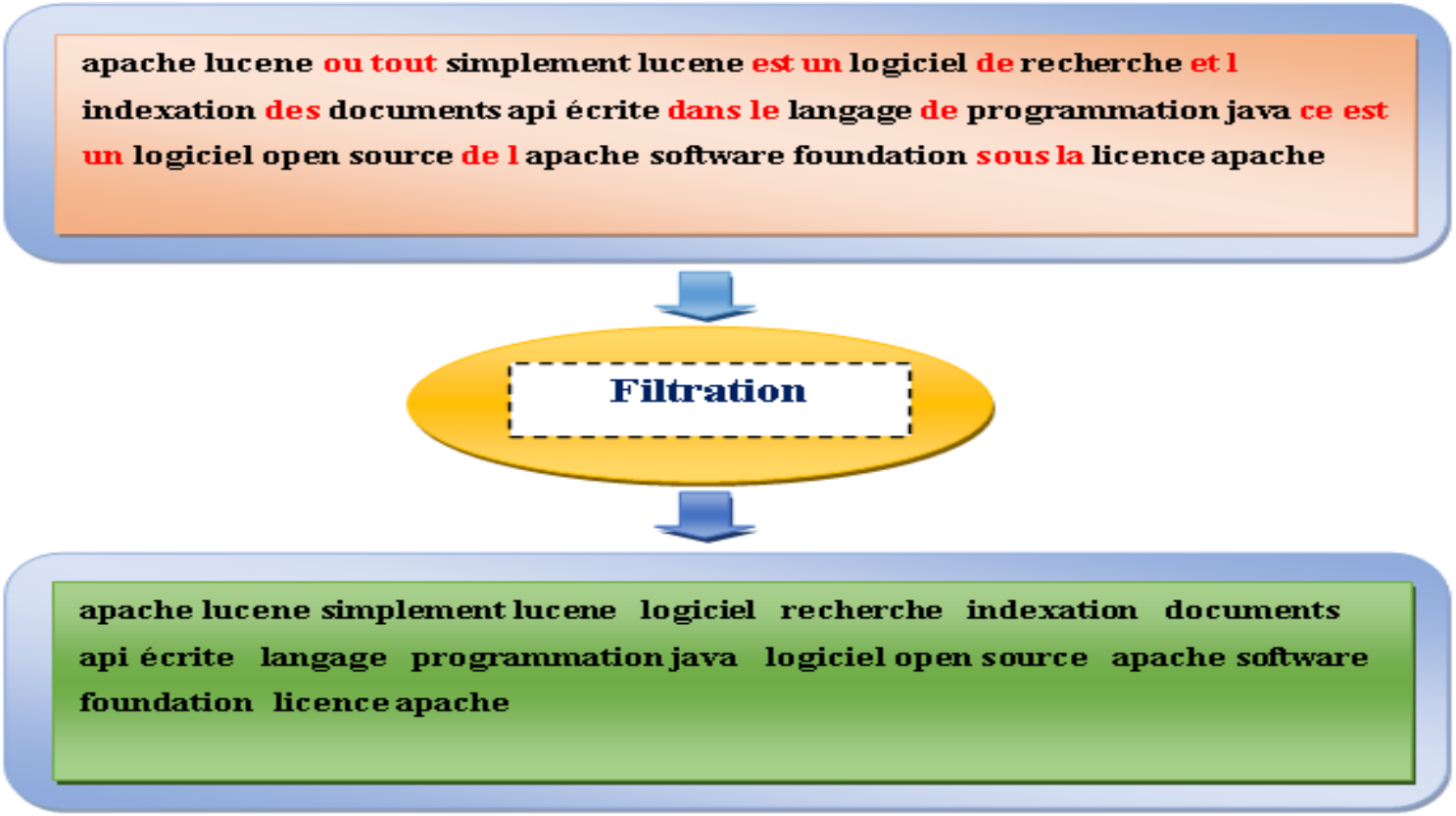
Variantes :

- WhitespaceAnalyzer Un analyseur très simple qui sépare juste la marque en
- utilisant l'espace blanc.
- StopAnalyzer Enlève les mots anglais communs et de liaison, inutiles pour l'indexation.
- SnowballAnalyzer Un analyseur expérimental intéressant qui travaille sur le racines (recherche sur la raining renvoie aussi rained, rain)
- FrenchAnalyzer Un analyseur pour le Français

Analyse des Documents avec Lucene



Analyse des Documents avec Lucene



Analyse des Documents avec Lucene

Lemmatisation

Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Analyse des Documents

- **Extraction des phrases**
- l'analyse grammaticale

  - ➔ «syntagme nominal»
    » ➔ «chien noir » ou « petit enfant ».
  - ➔ «syntagme verbal».
    » ➔ :"Ilyès aime l'informatique".

Analyse des Documents

- **Extraction des Mots composé**
- Un mot composé est constitué de deux ou plusieurs mots attachés pour former un nouveau mot

- Par exemple

➔ « base de donnée », « *pommes de terre* » « *computer science* »

Analyse des Documents

- **Extraction des entités nommées**

- Jim bought 300 shares of Acme Corp. in 2006.

➔ [Jim]$_{\text{Person}}$ bought 300 shares of [Acme Corp.]$_{\text{Organization}}$ in [2006]$_{\text{Time}}$.

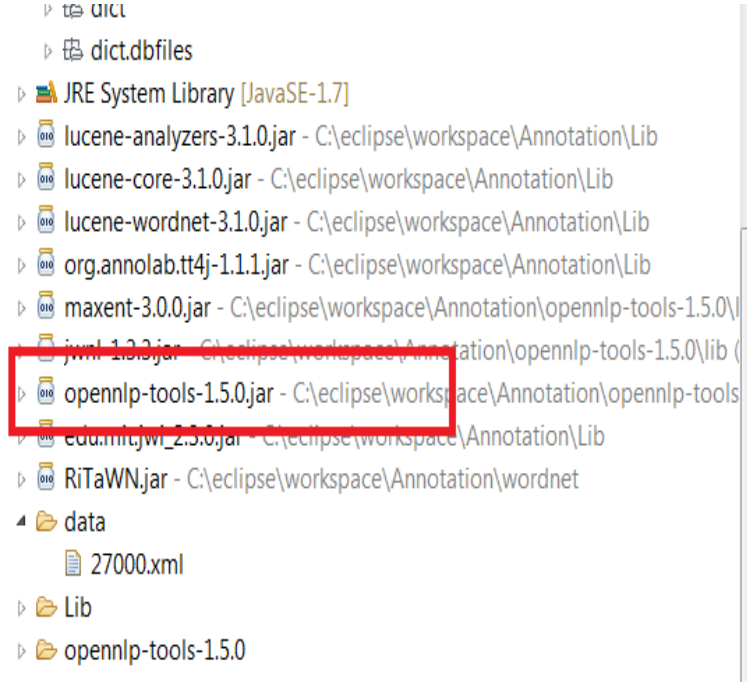
Analyse des Documents

- Unstructured Information Management Architecture (UIMA)
- General Architecture for Text Engineering (GATE)

- Apache OpenNLP library

Analyse des Documents

- **Apache OpenNLP library**




```
195         out2.println(txt2);
196         out.close();
197         out2.close();
198         return txtlist;
199     }
200 public static void main(String[] args) throws IOException {
201
202     NLP  nlp=new NLP();
203     String input=" ";
204     //String content="Pierre Vinken, 61 years old, will join the board as a nonexecut
205     nlp.tag(input);
206     //nlp.sent(content);
207     nlp.parse("los angeles is a state in USA");
208     //nlp.parse("Can anyone help me dig through OpenNLP's horrible documentation? ");
209     //nlp.parse("abdominal external oblique muscle");
210     //nlp.parse("external oblique muscle");
211
212 }
```

Analyse des Documents

- **Apache OpenNLP library**
- **Exemple :** Los Angeles is a state in USA

- (TOP (S (NP (NNS los) (NNS angeles)) (VP (VBZ is) (NP (DT a) (NN state)) (PP (IN in) (NP (NNP USA))))))

- NP los angeles
- NP a state
- NP USA

Analyse des Documents

Part of speech tags[1]

CC - Coordinating conjunction
CD - Cardinal number
DT - Determiner
EX - Existential there
FW - Foreign word
IN - Preposition or subordinating conjunction
JJ - Adjective
JJR - Adjective, comparative
JJS - Adjective, superlative
NN - Noun, singular or mass
NNS - Noun, plural
NNP - Proper noun, singular
NNPS - Proper noun, plural
PDT – Predeterminer
NP - Noun Phrase.
PP - Prepositional Phrase
VP - Verb Phrase.

PRP - Personal pronoun
RB - Adverb
RBR - Adverb, comparative
RBS - Adverb, superlative
RP - Particle
SYM - Symbol
TO - to
UH - Interjection
VB - Verb, base form
VBD - Verb, past tense
VBG - Verb, gerund or present participle
VBN - Verb, past participle
VBP - Verb, non-3rd person singular present
VBZ - Verb, 3rd person singular present
WDT - Wh-determiner
WP - Wh-pronoun
WRB - Wh-adverb

[1] http://bulba.sdsu.edu/jeanette/thesis/PennTags.html

Analyse des Documents

- **Extraction des concepts**
- Concept se concentre plus sur la sémantique du texte.
- C'est une unité abstraite de connaissances qui explique le sens des mots.
- Il faut utiliser les ressources sémantiques externes pour extraire le concept à partir du texte.

The noun computer science has 1 sense
1. computer science, computing -- (the branch of engineering science that studies (with the aid of computers) computable processes and structures)

Analyse des Documents

- **Extraction des concepts**

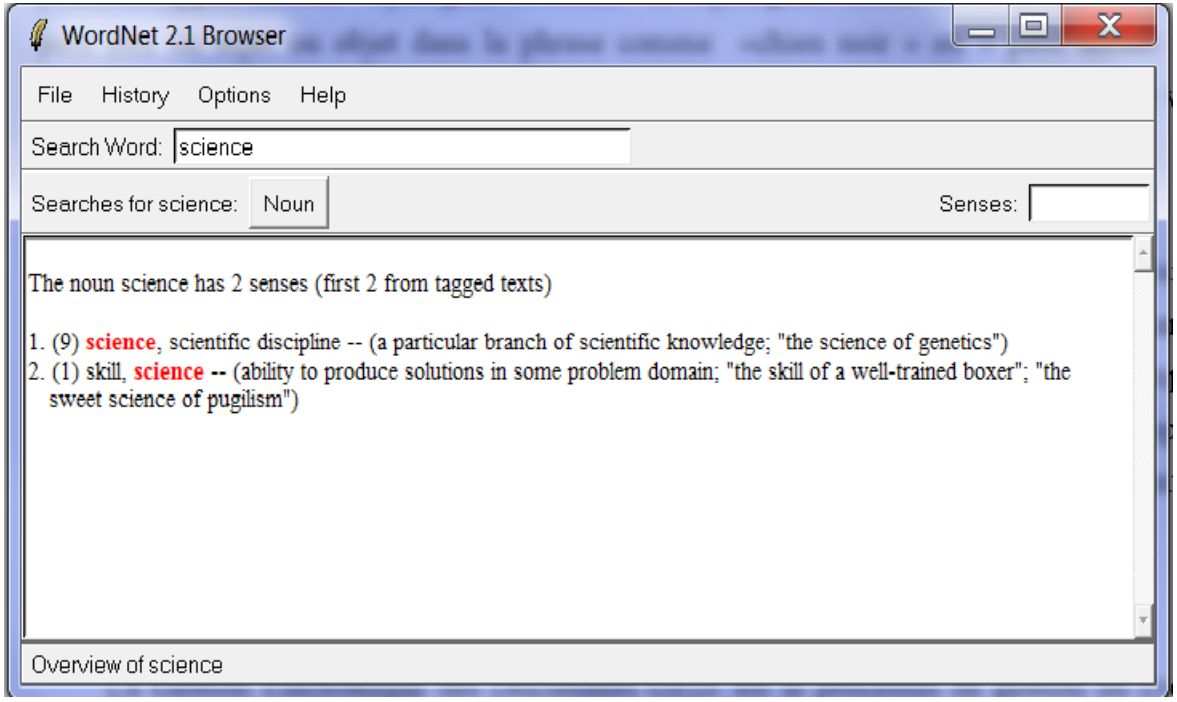
The noun **computer** has 2 senses (first 1 from tagged texts)

1. (6) computer, computing machine, computing device, data processor, electronic computer, information processing system -- (a machine for performing calculations automatically)

2. calculator, reckoner, figurer, estimator, computer -- (an expert at calculation (or at operating calculating machines))

Analyse des Documents

- **Extraction des concepts**



Indexation des Documents avec Lucene

- Document doc = new Document();
-
- Field contentField1=new Field("contenu", contenu, Field.Store.YES, Field.Index.ANALYZED);
- doc.add(contentField1);
-
- doc.add(new Field("Titre", Titre_principale, Field.Store.YES, Field.Index.ANALYZED));
-
- indexWriter.addDocument(doc);
- indexWriter.optimize();
- indexWriter.close();

Indexation des BDs avec Lucene

```
Void indexDocs(IndexWriter writer, Connection conn) throws Exception {
String sql = "select id, name from employe";
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery(sql);
while (rs.next()) {
Document d = new Document();
d.add(new Field("id", rs.getString("id"), Field.Store.YES,Field.Index.NO));
d.add(new Field("name", rs.getString("name"),
Field.Store.NO,Field.Index.TOKENIZED));
writer.addDocument(d);
}
}
```

Indexation des Documents avec Lucene

```
import java.util.logging.Logger;
import org.apache.lucene.analysis.standard.StandardAnalyzer;
import org.apache.lucene.document.Document;
import org.apache.lucene.document.Field;
import org.apache.lucene.index.IndexWriter;
import org.apache.lucene.store.FSDirectory;
import org.apache.lucene.util.Version;


public class Index {

    public void indexFile(String contenu, String Titre_principale) throws IOException {

        File indexDir=new File("C:/Users/pc/Documents/NetBeansProjects/Lucene/Index");

        IndexWriter indexWriter = new IndexWriter(FSDirectory.open(indexDir),new StandardAnalyzer(Version.LUCENE_30),false

        Document doc = new Document();


         Field contentField1=new Field("contenu", contenu, Field.Store.YES, Field.Index.ANALYZED);
          doc.add(contentField1);

        doc.add(new Field("Titre", Titre_principale, Field.Store.YES, Field.Index.ANALYZED));
         indexWriter.addDocument(doc);
          indexWriter.optimize();
          indexWriter.close();

        }
```

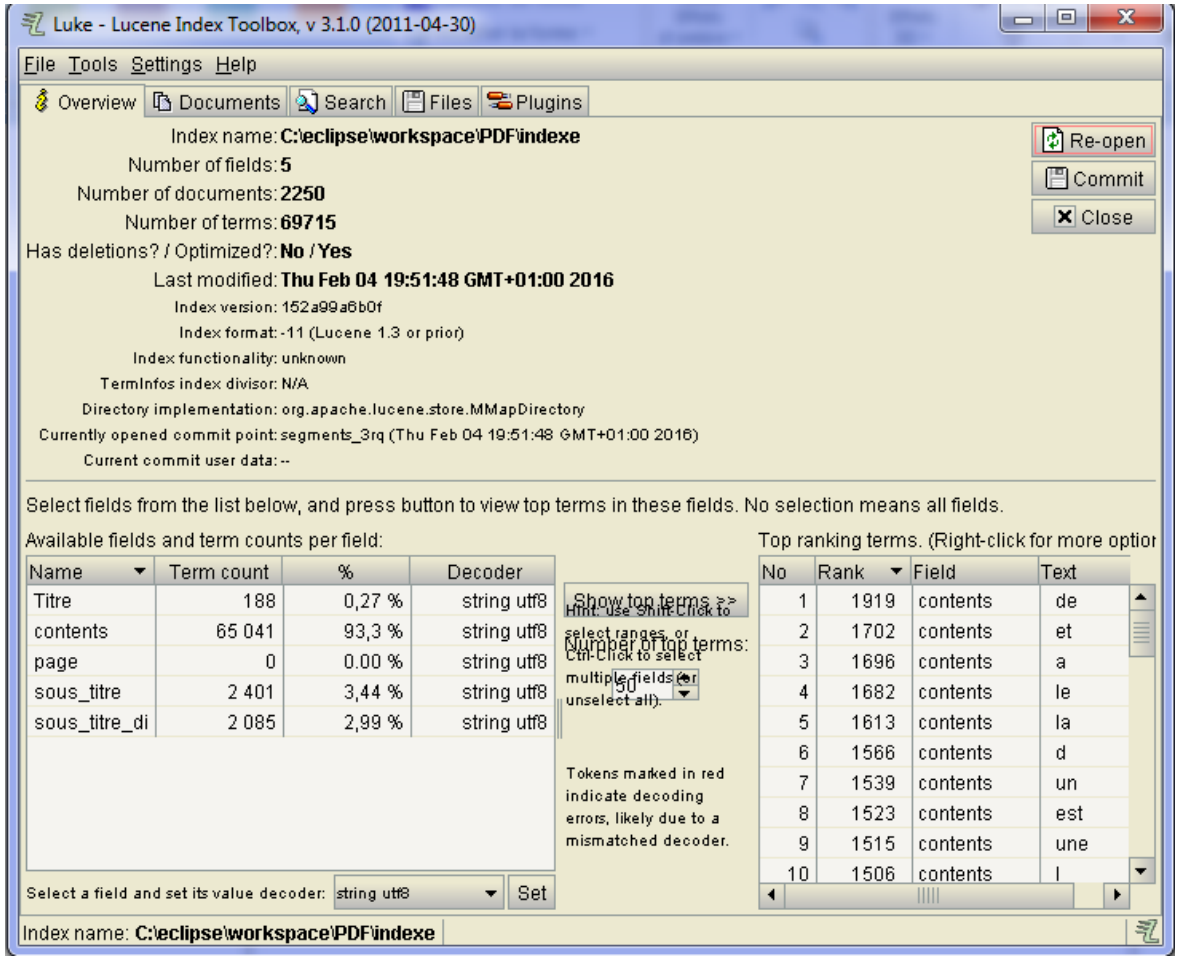
Indexation des Documents avec Lucene

```
    public static void main(String[] args) {
            // TODO Auto-generated method stub
        String contenu="nous allons ´etudier l'utilisation d'un index de recherche textuel OpenSource : Lucene";
        String titre="Lucene in Action";

        String contenu2="Pour aller plus loin, nous vous proposons de cr´eer un index pour un site web. Pour cela, v
"un objet qui va r´ecup´eer la page principale d'un site web d´etermin´e par son domaine. Ensuite pour chaque\n" +
"ancre (lien href) pr´esent sur la page, vous parcourez le lien s'il est sur le m^eme site web et le parcourir\n" +
"r´ecursivement3";
        String titre2="Recherche d'information";

         Index ndx =new Index();
        try {
            ndx.indexFile(contenu, titre);
```

Indexation des Documents avec Lucene

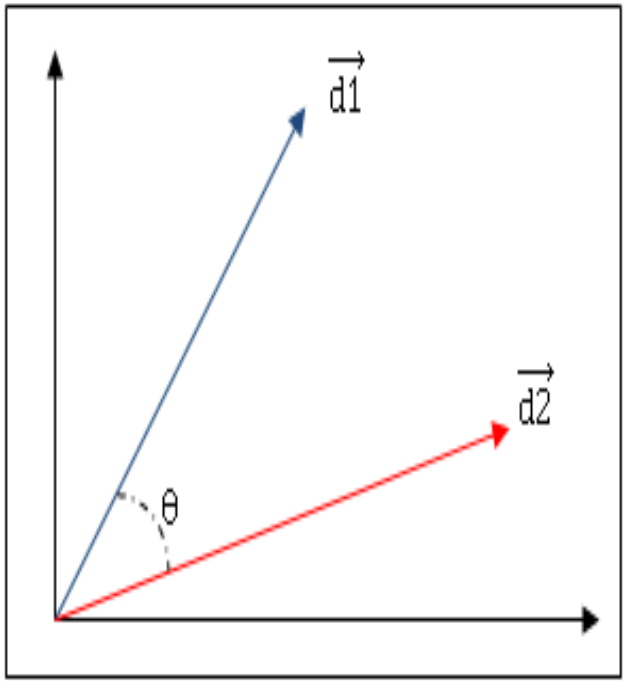
```
public class Recherche {

    public void search(File indexDir,String queryStr) throws IOException, ParseException{
        int i;
        Directory directory = FSDirectory.open(indexDir);
        IndexReader reader = IndexReader.open(directory);
        IndexSearcher searcher = new IndexSearcher(reader);
         QueryParser parser =new QueryParser(Version.LUCENE_31,"contenu", new StandardAnalyzer(Version.LUCENE_31));
                Query query = parser.parse(queryStr);

                TopDocs topDocs = searcher.search(query, 10);
                ScoreDoc[] hits = topDocs.scoreDocs;
                for ( i = 0; i < hits.length; i++) {

                                        int docId = hits[i].doc;
                                        Document d = searcher.doc(docId);

                                        System.out.println("-------------");
                                        System.out.println(i+1+"- " +d.get("Titre"));
                                }          }

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

        File indexDir = new File("C:/Users/pc/Documents/NetBeansProjects/Lucene/Index");
        Recherche R =new Recherche();
        R.search(indexDir, "site web");
    }

}
```

Indexation des Documents avec Lucene



Indexation des Documents avec Lucene

- Requête / Document
- La similarité entre deux vecteurs de documents (d1: requête) et (d2)



- $\text{Cos}(\theta) = \frac{\overrightarrow{d_1} \cdot \overrightarrow{d_2}}{|\overrightarrow{d_1}|\,|\overrightarrow{d_2}|}$
- Documents et requêtes sont représentés
- comme des vecteurs
- D1={w11, w21,....., w N, 1 },
- q= { w1,q, w2,q,....., w N, q }
- $\overrightarrow{d_1} \cdot \vec{q} = \sum wd1 * wq$
- Norme $|d_1| = \sqrt{w1^2 + w2^2 + w3^2}$ …..