Optativa: Análisis y Recuperación de Información

**Alumnos:**

**Mauricio Morales**

**Néstor Daniel Altamirano**

Profesora:

**Daniela Godoy**

Cursada 2013

# Incisos

1. Considere dos motores de búsqueda A y B que utilizan la misma colección de documentos. Cada motor de búsqueda retorna los 30 documentos mejor rankeados para cada búsqueda, ordenados por relevancia con respecto a la misma. Suponiendo que existen 12 documentos relevantes en la colección para la búsqueda Q, y que los resultados arrojados por ambos buscadores son los siguientes:

A: + + - + - - - + - - - + - - - - - - - + - - - - - - - - - +

B: - + + - - - + - - + + - + - - - + - - + - - + - + - - - - -

Donde + indica un documento relevante en la posición dada y - un documento irrelevante.

* 1. Dibujar el gráfico de precisión/recall para ambos motores.
  2. ¿Qué motor arroja mejores resultados? Explicar

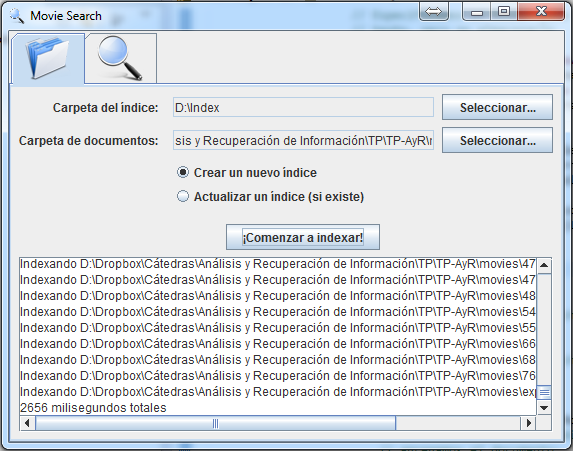
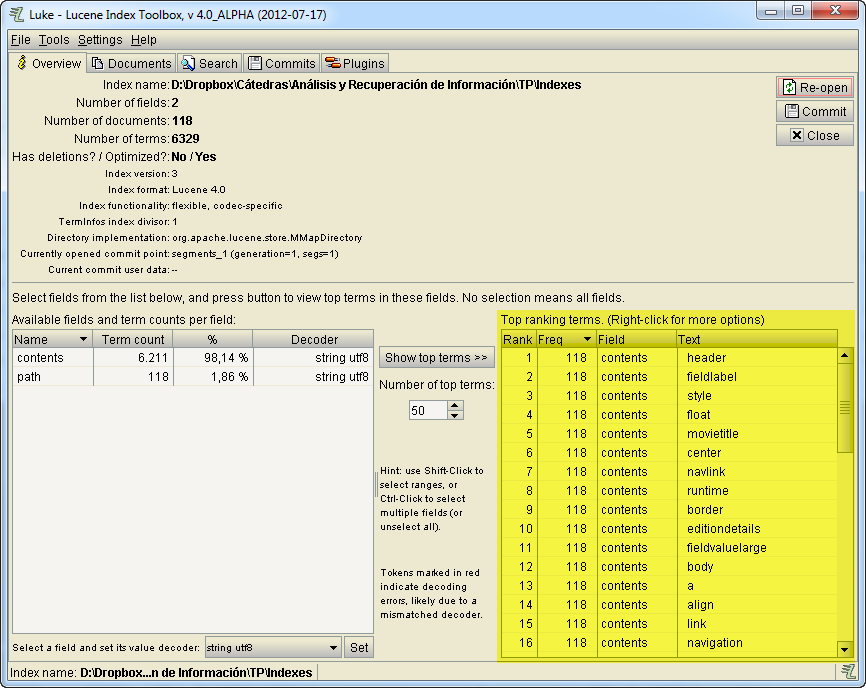
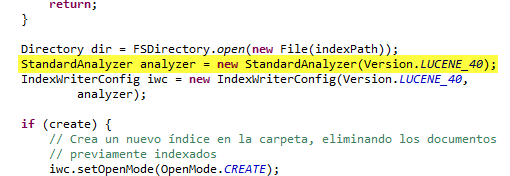
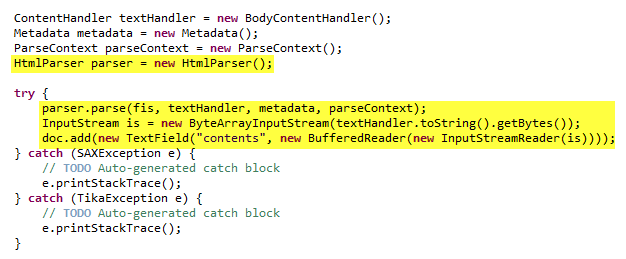
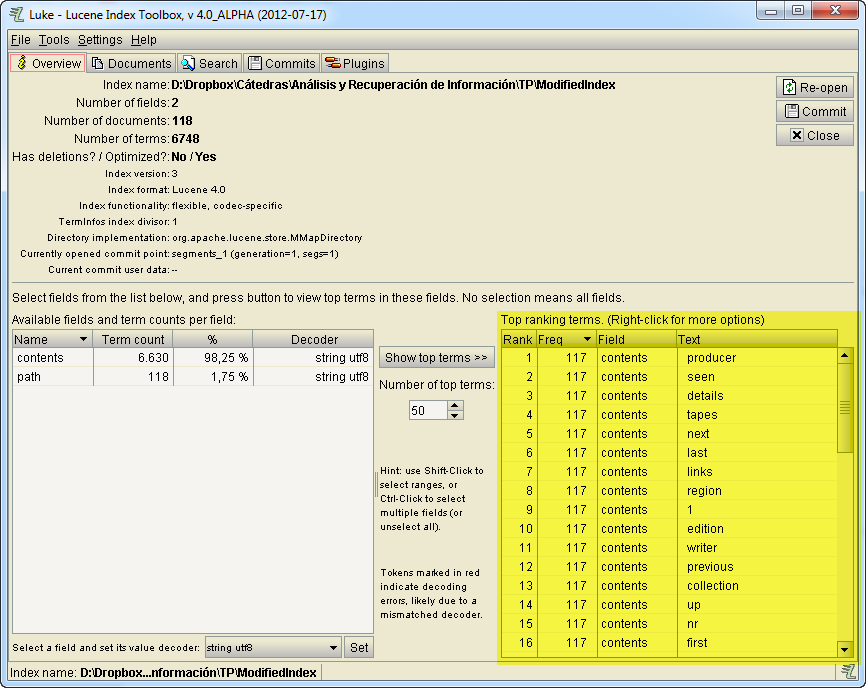
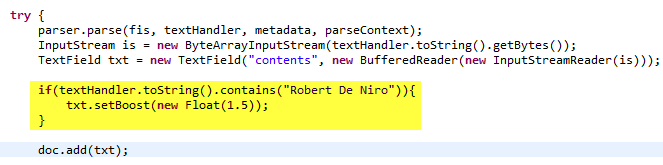
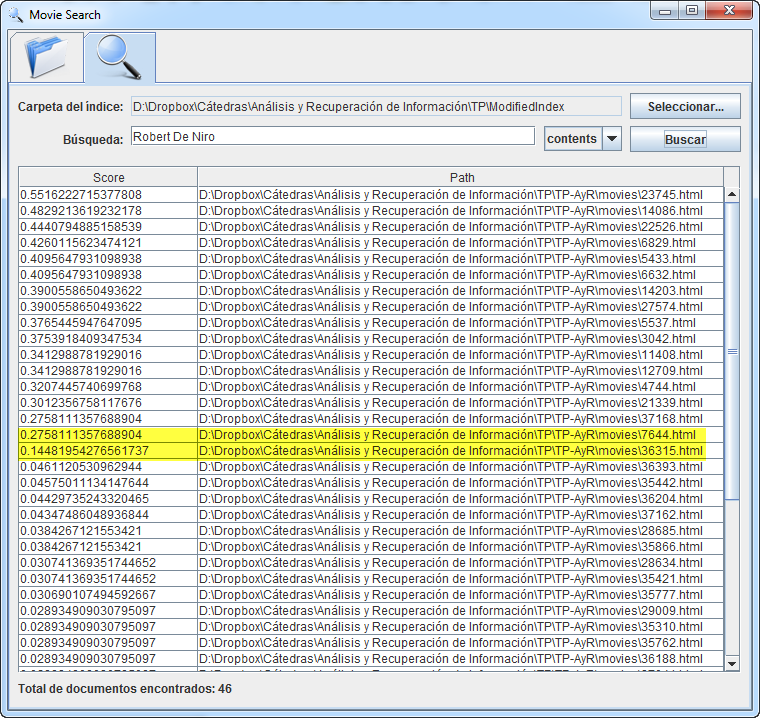
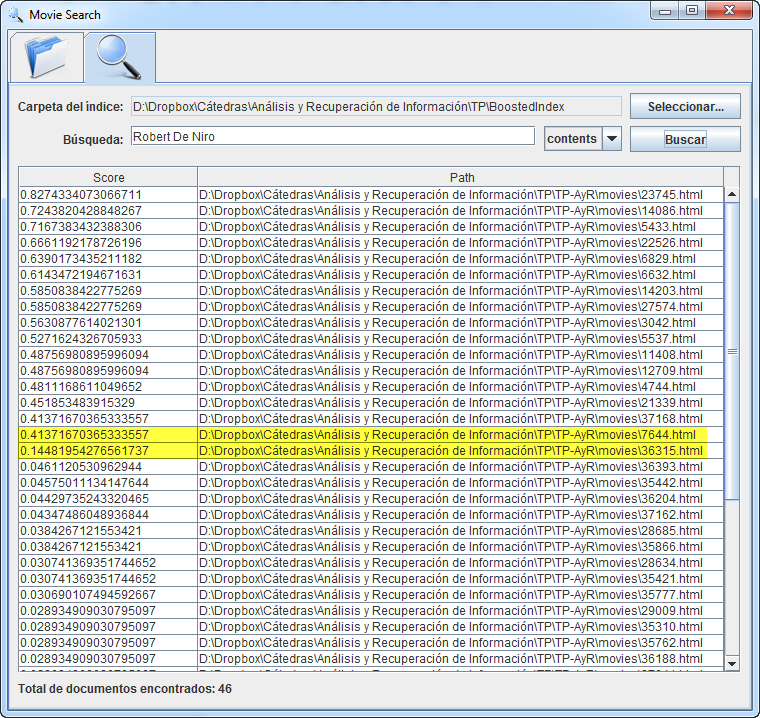
1. Utilizar la aplicación provista por la cátedra para indexar la carpeta "movies" provista.
2. Utilizar la aplicación luke de Apache, para inspeccionar el índice creado. Analizar si la información indexada es la esperada. En caso de no serlo, indicar cuál es el problema que presenta
3. Inspeccione el código provisto de la aplicación provista por la cátedra. Describa qué cambiaría para solucionar el problema presentado en el punto 3.
4. ¿Qué modificaría de la aplicación provista para que las películas en las que participa *Robert De Niro* aparezcan mejor posicionadas en el ranking de resultados?
5. ¿Cómo modificaría la aplicación para realizar una búsqueda avanzada por género, actores y director?

# Resolución

* 1. Relevantes: 12 documentos  
     Recuperados: 30 documentos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | | | B | | |
| **Resultados** | **Aciertos** | **Recall** | **Precisión** | **Aciertos** | **Recall** | **Precisión** |
| 1 | 1 | 0,08 | 1,00 | 0 | 0,00 | 0,00 |
| 2 | 2 | 0,17 | 1,00 | 1 | 0,08 | 0,50 |
| 3 | 2 | 0,17 | 0,67 | 2 | 0,17 | 0,67 |
| 4 | 3 | 0,25 | 0,75 | 2 | 0,17 | 0,50 |
| 5 | 3 | 0,25 | 0,60 | 2 | 0,17 | 0,40 |
| 6 | 3 | 0,25 | 0,50 | 2 | 0,17 | 0,33 |
| 7 | 4 | 0,33 | 0,57 | 3 | 0,25 | 0,43 |
| 8 | 4 | 0,33 | 0,50 | 3 | 0,25 | 0,38 |
| 9 | 4 | 0,33 | 0,44 | 3 | 0,25 | 0,33 |
| 10 | 4 | 0,33 | 0,40 | 4 | 0,33 | 0,40 |
| 11 | 5 | 0,42 | 0,45 | 5 | 0,42 | 0,45 |
| 12 | 5 | 0,42 | 0,42 | 5 | 0,42 | 0,42 |
| 13 | 5 | 0,42 | 0,38 | 6 | 0,50 | 0,46 |
| 14 | 5 | 0,42 | 0,36 | 6 | 0,50 | 0,43 |
| 15 | 5 | 0,42 | 0,33 | 6 | 0,50 | 0,40 |
| 16 | 5 | 0,42 | 0,31 | 6 | 0,50 | 0,38 |
| 17 | 5 | 0,42 | 0,29 | 7 | 0,58 | 0,41 |
| 18 | 5 | 0,42 | 0,28 | 7 | 0,58 | 0,39 |
| 19 | 6 | 0,50 | 0,32 | 7 | 0,58 | 0,37 |
| 20 | 6 | 0,50 | 0,30 | 8 | 0,67 | 0,40 |
| 21 | 6 | 0,50 | 0,29 | 8 | 0,67 | 0,38 |
| 22 | 6 | 0,50 | 0,27 | 8 | 0,67 | 0,36 |
| 23 | 6 | 0,50 | 0,26 | 9 | 0,75 | 0,39 |
| 24 | 6 | 0,50 | 0,25 | 9 | 0,75 | 0,38 |
| 25 | 6 | 0,50 | 0,24 | 10 | 0,83 | 0,40 |
| 26 | 6 | 0,50 | 0,23 | 10 | 0,83 | 0,38 |
| 27 | 6 | 0,50 | 0,22 | 10 | 0,83 | 0,37 |
| 28 | 6 | 0,50 | 0,21 | 10 | 0,83 | 0,36 |
| 29 | 6 | 0,50 | 0,21 | 10 | 0,83 | 0,34 |
| 30 | 7 | 0,58 | 0,23 | 10 | 0,83 | 0,33 |

* 1. El motor B arroja mejores resultados, ya que la precisión termina siendo del 33% de resultados relevantes sobre el total de resultados relevantes posibles, mientras que la del motor A este valor baja al 23%. Además, la recuperación de resultados relevantes para el motor B es del 83% de resultados relevantes sobre el total de resultados obtenidos del conjunto, mientras que en el motor A, este indicador es del 58%.

1. Indexamos la carpeta Movies con la aplicación provista por la cátedra.  
   
2. El índice creado está indexando tags HTML que no debería, y además no excluye StopWords.  
   
3. A fin de excluir StopWords del índice, utilizamos un StandardAnalyzer en lugar de un SimpleAnalyzer, el cual incluye un conjunto de StopWords a filtrar por defecto.  
     
     
   Luego, para excluir los tags HTML, utilizamos la librería Tika. Creamos un parser de tipo HtmlParser, y utilizamos el texto extraído para agregarlo al documento, sin incluir los tags HTML.  
     
     
     
   Y el resultado son las palabras importantes dentro del texto, sin incluir tags ni StopWords.  
   
4. A fin de mejorar el posicionamiento de las películas en las que participe Robert De Niro, podemos aumentar el *Boosting* de aquellos documentos que contengan su nombre en el texto.  
     
     
     
   De esta manera, podemos observar la diferencia entre el score de una película en la que participa Robert De Niro (archivo **7644.html**) versus el score de una película en la que no participa (archivo **36315.html**), antes y después de realizar el cambio en el código.  
     
   **Índice antes de aumentar el *Boosting***  
     
   **Índice luego de aumentar el *Boosting***  
   
5. A fin de poder realizar una búsqueda avanzada por género, actores y director, lo que haríamos es parsear previamente la información al procesar cada página, extrayendo esos tres datos por separado, y almacenarlos en campos separados del contenido principal.  
     
   Para lograr esto, podemos utilizar expresiones regulares que levanten la información del texto de cada página, y luego de separarla, crear campos *Genero*, *Actores*, y *Director*.