

**PROCESADO DE DATOS**

En primer lugar, la tarea a realizar era procesar una serie de segmentos obtenidos del crawler configurado en la anterior práctica llamado “Apache Nutch”. El crawler procesaba una serie de webs en las que estaban almacenados los documentos de trabajos fin de grado y fin de máster a indexar por nuestro sistema tradicional. La idea principal era, a partir del programa para indexar los metadatos en formato “xml”, ampliarlo para que fuera capaz de hacer lo mismo con los segmentos del crawler que contenían en su interior dichos metadatos.

Quedando clara la idea, se procedió a obtener los segmentos con el uso del crawler configurado en la práctica anterior. El crawler estaba compuesto por las etapas de inyección de URLs (añade las URLs a la colección de las que se deben procesar), y la etapa de “Fetch”, en la que se accede a las URLs de la lista y recupera su contenido.

Las etapas se podían ejecutar de una en una, obteniendo de cada una de ellas un segmento, y dejando en la lista de nuevas URLs a procesar las obtenidas en la información de dicho segmento, o mediante un script proporcionado por “Apache Nutch”, que realizaba todas las fases seguidas, sin necesidad de pasar manualmente una por una. Para ejecutar el Script se utilizó el siguiente comando:

“bin/crawl urls/seed.txt MiCrawl <http://localhost:8983/solr> 10”

El directorio que contenía las semillas o primeras URLs con las que empezar el proceso de crawling era “urls/seed.txt”. “MiCrawl” era el identificador definido para nuestro crawler y que iba a ser el encargado de descargar la web de una cierta dirección. En nuestro caso, dicha dirección era “<http://localhost:8983/solr>”, y a la que había que agregar un cierto número que marcaría el número de iteraciones necesarias para descargar todo lo contenido en la URL dada. En un principio, pensamos en indicarle 3 o 4 iteraciones ya que creíamos que serían suficiente, pero finalmente optamos por 10 a pesar de que finalmente observamos que se generaban solo 5 segmentos.

Una vez obtenidos los 5 segmentos, procedimos a analizarlos para saber su estructura y cómo podrían ser indexados. Al verlos, nos dimos cuenta que aún faltaba un paso más, que era volcarlos para obtener la información relevante, ya que los segmentos en sí no se podían analizar. Así pues, procedimos a volcar los segmentos con el siguiente comando:

“bin/nutch readseg -dump MiCrawl/segments/<id del segmento> dirVolcado”

Ejecutando este mismo comando para los 5 segmentos generados por el proceso de crawling, obtuvimos ya la información final que iba a ser indexada. Así pues, se procedió a realizar un análisis de lo que contenía cada archivo obtenido de volcar los segmentos, y a diseñar un proceso eficiente para que fueran indexador por nuestro programa creado en tareas anteriores.

Cada segmento tenía la información relacionada con un cierto subdirectorio relacionado con la URL dada. Así pues, el primero de ellos contenía las URLS de los 4 subdirectorios, además de otra información como URLS seleccionadas y procesadas… (se quitó bastante información que no era relevante añadiendo distintas opciones al comando para volcar los segmentos). Los otros 4 segmentos obtenidos ya sí que contenían los metadatos de los trabajos fin de grado y carrera en formato “xml”, pero también seguían llevando cierta información extra que no resultaba de interés.

Gracias al análisis realizado de los archivos a indexar, se diseñó un proceso para parsear los archivos, extrayendo la información relevante (metadatos de trabajos “xml”) e ignorando aquella que no aportaba información. Además, este proceso se realizó ahorrando un gran número de líneas de código, y reutilizando lo diseñado en prácticas anteriores como se describe en el siguiente apartado.

**INDEXACIÓN**

A partir del volcado de los segmentos, y del análisis de los archivos generados, se procedió a la indexación. El proceso a seguir era, en primer lugar, recorrer el directorio con los segmentos, leyendo uno a uno cada uno de ellos. Este segmento se leía línea por línea hasta encontrar algo que cumplía un cierto patrón de formato XML. La información que cumplía el patrón, se extraía entera (ya fuera una sola línea o varias), y se guardaba en un fichero temporal.

Si se miraba la información que se extraía y se escribía en el fichero temporal, se podía apreciar que era un documento exactamente igual al que se proporcionaba para indexar en nuestro sistema tradicional diseñado anteriormente. Así pues, sólo había que parsearlo en los distintos campos como ya se había hecho anteriormente, por lo que se podía pasar dicho fichero al parser de prácticas anteriores, y él realizaba el proceso necesario para extraer los distintos campos y su información, y finalmente, indexándolo en “Lucene”.

Se analizó otros procedimientos como ir indexando directamente el contenido XML reconocido de los segmentos, pero se decidió que este era el más adecuado. Esto es debido a que, con la creación de los ficheros temporales, que al finalizar el proceso se eliminaban, se podía reutilizar el código del sistema tradicional para realizar el parser e indexación, centrando sólo los esfuerzos en sacar la información de los metadatos de los segmentos, y pasándosela al código que ya estaba diseñado para el sistema tradicional.

**SISTEMA TRADICIONAL VS SOLR**

Finalmente, tras haber conseguido de forma satisfactoria indexar la información contenida en los segmentos creados por nuestro crawler, se procedió a realizar una comparativa entre los resultados obtenidos por nuestro proceso de búsqueda, frente a la configuración por defecto de “Solr”. Para ello, en primer lugar, era necesario indexar nuestros segmentos en “Solr”, tarea compuesta por dos procesos. El primero de ellos era realizar un ranking basado en la relevancia de las páginas, relacionado con el número de páginas web que apuntan a una cierta dada:

“bin/nutch invertlinks MiCrawl/linkdb -dir MiCrawl/segments”

Una vez obtenido el ranking por relevancia, se procedió a indexar los segmentos en “Solr” con el siguiente comando:

“bin/nucth solrindex <http://127.0.0.1:8983/solr/> MiCrawl/crawldb -linkdb MiCrawl/linkdb MiCrawl/segments/\*”

A continuación, se realizó una comparación entre los resultados obtenidos por nuestro sistema tradicional, frente a los dados por “Solr”. Para ello, se seleccionó un conjunto de palabras de la necesidad de información y se realizó la búsqueda en “Solr”.

Consulta 01-1

Para esta consulta, se seleccionaron las palabras “geológica” y “España”, realizando la siguiente búsqueda en “Solr”:

“http://localhost:8983/solr/select?q=geológica+España”

La tabla comparativa con los resultados es la siguiente:

|  |  |
| --- | --- |
| **Sistema Tradicional** | **Solr** |
| oai\_zaguan.unizar.es\_13324.xml | oai\_zaguan.unizar.es\_56632.xml |
| oai\_zaguan.unizar.es\_9709.xml | oai\_zaguan.unizar.es\_56633.xml |
| ai\_zaguan.unizar.es\_31157.xml | oai\_zaguan.unizar.es\_56714.xml |

Como se puede ver, ninguno de los tres primeros documentos del ranking coincide con los del otro sistema. Mientras que los del sistema tradicional cuadran con la búsqueda realizada, el primero de “Solr” habla sobre la violencia de género en España, por lo que se ofrecen mejores resultados en el tradicional.

Consulta 04-3

Para esta consulta, se seleccionaron las palabras “enfermedades”, “infecciosas”, “1980” y “1990”, realizando la siguiente búsqueda en “Solr”:

“http://localhost:8983/solr/select?q=enfermedades+infecciosas+1980+1990”

|  |  |
| --- | --- |
| **Sistema Tradicional** | **Solr** |
| oai\_zaguan.unizar.es\_10561.xml | oai\_zaguan.unizar.es\_56710.xml |
| oai\_zaguan.unizar.es\_31772.xml | oai\_zaguan.unizar.es\_56684.xml |
| oai\_zaguan.unizar.es\_7108.xml | oai\_zaguan.unizar.es\_56715.xml |

De igual modo que en la anterior consulta, los resultados del sistema tradicional cuadran con el tema “enfermedades infecciosas”, aunque la fecha en algunos casos no esté en el rango fijado. Sin embargo, nuevamente “Solr” acierta menos, aunque en este caso se acerca más ya que el primer resultado habla sobre enfermedades, aunque sean “celíacas”.

Consulta 05-1

Para esta consulta, se seleccionaron las palabras “algoritmos”, “modelos”, “funciones”, “geométricas” e “implementación”, realizando la siguiente búsqueda en “Solr”:

“http://localhost:8983/solr/select?q=algoritm+modelos+funciones+geometricas+implementacion”

|  |  |
| --- | --- |
| **Sistema Tradicional** | **Solr** |
| oai\_zaguan.unizar.es\_15314.xml | oai\_zaguan.unizar.es\_56718.xml |
| oai\_zaguan.unizar.es\_8916.xml | oai\_zaguan.unizar.es\_56714.xml |
| oai\_zaguan.unizar.es\_4977.xml | oai\_zaguan.unizar.es\_56716.xml |

Nuevamente, el primer resultado del sistema tradicional clava la consulta, mientras que, por ejemplo, el primer resultado de “Solr” habla sobre enfermedades neuro-generativas. El motivo de que haya salido tan arriba en el ranking este documento es que contiene varias veces la palabra “funciones”.

Consulta 09-4

Para esta consulta, se seleccionaron las palabras “robótica”, “2005”, “Martínez”, realizando la siguiente búsqueda en “Solr”:

“http://localhost:8983/solr/select?q=robótica+2005+Martinez”

|  |  |
| --- | --- |
| **Sistema Tradicional** | **Solr** |
| oai\_zaguan.unizar.es\_16912.xml | oai\_zaguan.unizar.es\_56710.xml |
| oai\_zaguan.unizar.es\_4498.xml | oai\_zaguan.unizar.es\_56692.xml |
| oai\_zaguan.unizar.es\_47945.xml | oai\_zaguan.unizar.es\_56472.xml |

En esta consulta ambos sistemas dan resultados erróneos. Mientras que las dos primeras del sistema tradicional no tienen nada que ver con el tema, hablando la primera sobre poesía, la de “Solr” vuelve a hablar sobre enfermedades celíacas probablemente porque fue realizado por una persona de apellido “Martínez”.

Consulta 11-5

“http://localhost:8983/solr/select?q=inteligencia+artificial+videojuego+ultimos+5+años”

|  |  |
| --- | --- |
| **Sistema Tradicional** | **Solr** |
| oai\_zaguan.unizar.es\_4876.xml | oai\_zaguan.unizar.es\_56692.xml |
| oai\_zaguan.unizar.es\_31467.xml | oai\_zaguan.unizar.es\_56683.xml |
| oai\_zaguan.unizar.es\_8010.xml | oai\_zaguan.unizar.es\_56711.xml |

Finalmente, en esta consulta, como de costumbre, el sistema tradicional da los resultados perfectos en sus tres primeras posiciones, mientras que nuevamente “Solr” no da una, colocando en la primera posición de su ranking un trabajo sobre la obesidad.