Trabajo MiniTREC: Entrega de los resultados del crawler

Guillermo Cruz (682433) Francisco Ferraz (737312)

Diferencia entre procesos de indexación y búsqueda en el sistema tradicional y SOLR

En primer lugar, el sistema tradicional, que crea los índices con el software Lucene, trabaja con metadatos Dublin Core, y se puede tener índices separados para cada campo o elemento de los metadatos (title, identifier, subject, etc.) especificándolo en la clase IndexFiles, en vez de tener un único índice para todo el contenido. Así que por ello en el sistema tradicional se consigue tener indexado en diferentes índices el contenido de cada uno de los campos, lo que será de utilidad para realizar las consultas como se mostrará más adelante.

En SOLR, la indexación funciona de forma diferente. Una vez que el crawler ha descargado todas las páginas de la colección y procesado su contenido, es la hora para SOLR de indexarlo. A pesar de que las páginas siguen teniendo el esquema de metadatos Dublin Core, ya que es la misma colección, este sistema no tendrá campos diferentes como subject, description, etc. sino que tendrá el contenido en texto por completo de la página web en cuestión (campo contents), no diferencia entre campos, por lo que no se puede realizar un índice diferente para cada uno de ellos (al menos en esta versión por defecto) como se podía hacer en Lucene. A continuación se explica cómo se ha llegado a esta conclusión.

Mientras que en el sistema tradicional se indica cómo indexar cada campo en la clase IndexFiles, en SOLR se especifica en el fichero schema.xml (example/solr/collection1/conf/schema.xml). Es necesario cambiar este fichero para modificar el esquema de metadatos definido por defecto en SOLR por el usado por Nutch para representar los documentos descargados por el crawler.

Se puede observar como en la sección <types>, se definen los diferentes tipos de datos que maneja SOLR para indexar los datos. Sería el equivalente a los datos tipo StringField, TextField, LongPoint, etc. de Lucene. Los más relevantes para nuestra colección son date, text_general (un campo para textos que se tokeniza con StandardTokenizer) y url. Cada uno de estos tipos de dato son también los que indican el procesado que se le realiza al contenido cuando se indexa, como no indexar las stopwords, aplicar sinónimos, etc. También procesan el texto de la consulta.

A continuación, en la sección <fields>, se señalan los campos de los metadatos descargados que se quiere que sean almacenados e indexados. Los más importantes en este esquema, son url, content y title. Para saber cuáles son los campos que el crawler descargaba, para luego poder indexarlos, se ha utilizado la herramienta de volcado en texto del contenido de un segmento que se proporciona en la práctica 4 mediante el comando:

bin/nutch readseg -dump micrawl/segments/<id seg> dirVolcado

En el fichero generado con toda la información, se ha podido observar que en el campo content se almacena todo el contenido de la url tal cual está, es decir todo el texto sin realizar ningún tipo de parseado, pero más tarde se ha observado que lo que

SOLR almacena e indexa es el texto habiendo parseado y eliminado todas las etiquetas del tipo <dc:___>, es decir el campo ParseText del fichero de volcado. A continuación se muestra un ejemplo en la imagen 1:

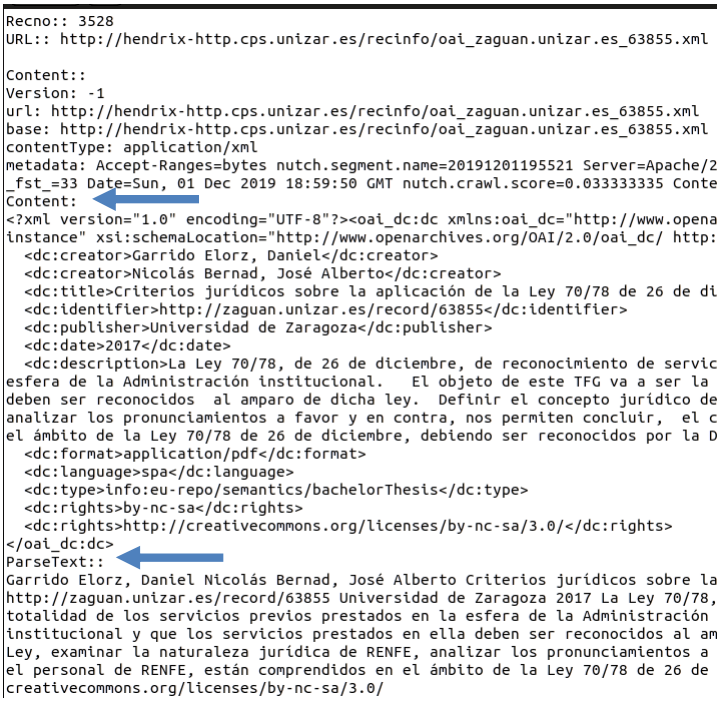

Imagen 1. Campo content y ParseText almacenado por el crawler

Cabe destacar, que en las páginas que representan directamente un trabajo (no una categoría), el campo title queda vacío, por lo que la única manera de buscar información es realizando consultas sobre el campo content, que como ya se ha dicho guardará todo el texto encontrado en la página en un bloque, incluyendo autores, título, año, etc., por lo que con la configuración por defecto no hay manera de restringir una búsqueda sobre un campo específico de nuestra colección. Es decir, la consulta se ejecutará sobre todo el texto de la página en cuestión, al contrario que en el sistema tradicional, que si se puede.

Antes de realizar ningún experimento, se puede apreciar que los resultados de las consultas que se ejecuten en SOLR pueden ser diferentes a los del sistema tradicional, ya que si con el sistema tradicional se puede buscar un nombre específico en el campo autor, en SOLR nos devolverá las páginas que contengan dicho nombre en cualquier lugar.

Resultados

Se ha procedido a realizar las consultas apropiadas para cada necesidad de información en cada sistema. Cabe destacar que el sistema tradicional, a la hora de realizar las consultas, aplica el algoritmo de stemming de Snowball para el idioma español gracias al analizador empleado, *SpanishAnalyzer.* Por lo tanto se busca la raíz de la palabra. Algo similar se puede utilizar al buscar en SOLR, utilizando el operador '~' seguido de un número entre 0 y 1. Buscará palabras semejantes en el rango 0-1, con un 1 para máxima igualdad entre palabras. Con el operador '^' se le da más importancia a esa palabra a la hora de generar el ranking. Se muestran los 5 primeros resultados.

No se pueden realizar consultas de rangos numéricos, ya que no hay un campo date, por lo que como se ha descubierto que la fecha de publicación iba siempre precedida de la palabra Zaragoza, por ello se han realizado búsquedas del tipo *content:"Zaragoza 2013"*

Necesidad 1:

Consulta sistema tradicional:

date:[20100000 TO 20150000] AND (title:enfermedad title:ocular description:enfermedad description:ocular subject:ocular)

Consulta SOLR:

content:ocular^4 content:enfermedad AND (content:"Zaragoza 2010" content:"Zaragoza 2011" content:"Zaragoza 2012" content:"Zaragoza 2013" content:"Zaragoza 2014" content:"Zaragoza 2015")

Resultados tradicional	Resultados SOLR
9904, 11449, 47270, 31706, 12684	48030, 47796, 47922, 47611

Necesidad 2:

Consulta sistema tradicional:

title:"siglo XX" description:"siglo XX" subject:"siglo XX" title:cine description:cine subject:cine title:ideologia description:ideologia subject:ideologia

Consulta SOLR:

content:"siglo XX"^4 content:ideologia~ content:cine^3

Resultados tradicional	Resultados SOLR
9979, 65005, 13688, 76721, 31292	78517, 78601, 78488, 78811, 62681

Necesidad 3:

Consulta sistema tradicional:

date:[20110000 TO 20190000] AND (title:"inteligencia artificial" description:"inteligencia artificial" subject:"inteligencia artificial" title:videojuegos description:videojuegos subject:videojuegos title:personaje description:personaje subject:personaje)

Consulta SOLR:

content:"inteligencia artificial" content:"desarrollo videojuegos"~10 content:"diseño personajes"~10 AND (content:"Zaragoza 2019" content:"Zaragoza 2018" content:"Zaragoza 2017" content:"Zaragoza 2016" content:"Zaragoza 2015"

content:"Zaragoza 2014" content:"Zaragoza 2013" content:"Zaragoza 2012" content:"Zaragoza 2011")

Resultados tradicional	Resultados SOLR
6493, 12373, 10648, 61390, 16068	62791, 12373, 69866, 9435, 69901

Necesidad 4:

Consulta sistema tradicional:

(title:contamina subject:contamina description:contamina) AND (title:España description:España subject:España title:Aragón description:Aragón subject:Aragón title:Zaragoza title:Zaragoza title:Zaragoza)

Consulta SOLR:

content:contaminacion~ AND (content:España content:Aragon content:Zaragoza)

Resultados tradicional	Resultados SOLR
65459, 76432, 47082, 32728, 5733	63660, 78615, 78471, 17183, 47537

Necesidad 5:

Consulta sistema tradicional:

date:[20120000 TO 20200000] AND type:bachelorthesis AND creator:Javier AND (title:informatica description:informatica subject:informatica)

Consulta SOLR:

content:Javier AND content:bachelorThesis AND content:informatica~ AND (content:"Zaragoza 2019" content:"Zaragoza 2018" content:"Zaragoza 2017" content:"Zaragoza 2016" content:"Zaragoza 2015" content:"Zaragoza 2014" content:"Zaragoza 2013" content:"Zaragoza 2012")

Resultados tradicional	Resultados SOLR
65350, 61127, 31579, 6801, 37090	62618, 47560, 15973, 61127, 61386

Conclusión

En cada una de las necesidades, se han recogido resultados mucho más relevantes con el sistema tradicional que con SOLR. Se debe, claramente, a que en el primero se pueden hacer búsquedas por campos, y en el segundo se realiza la búsqueda sobre todo el contenido de la página. A pesar de ello, SOLR ha funcionado bastante bien y realizando la consulta adecuada se pueden conseguir documentos relevantes.