MOTOR DE BÚSQUEDA

Recuperación y Acceso a la Información

Carlos Contreras Sanz 100303562

Miguel Xoel García Balsa 100291036

ÍNDICE DE CONTENIDOS

INDIZACIÓN DE DOCUMENTOS 1

- Indización de documentos Web
  - Los documentos son importados mediante la clase Documentos();
  - Se crean dos DBCollection (objetos de Mongo) para guardar diccionario y entidades.
  - Sobre estas colecciones, se crean los índices de Mongo sobre los campos por lo que insertamos. En este caso, *palabra* y *entidad.*

```
// PASO 3: Obtenemos las colecciones para trabajar con ellas
DBCollection coleccionDiccionario = db.getCollection("Diccionario");
DBCollection coleccionEntidades = db.getCollection("Entidades");

//PASO 4: Creamos los indices sobre los campos por los que insertaremos
db.getCollection("Diccionario").createIndex("palabra");
db.getCollection("Entidades").createIndex("entidad");
```

INDIZACIÓN DE DOCUMENTOS 2

- Usamos 4 hilos para agilizar el proceso de indización.
  - Siempre hay 4 hilos trabajando.
  - Cada hilo recibe un documento y limpia su contenido.
  - Inserta cada palabra en la base de datos sobre los campos anteriormente especificados, usando un cerrojo para evitar repeticiones.

```
BasicDBObject query = new BasicDBObject("entidad", entidadesDoc.get(i));
DBObject updated = new BasicDBObject().append("$set", new BasicDBObject().
        append("documento", nombreDocumento));

synchronized(coleccionEntidades){

    coleccionEntidades.update(query, updated, true, false);
```

MÓDULO I: EXPANSIÓN DE CONSULTA 1

- Mediante la clase Sinónimos se usa la base de datos de *WordNet*.
- Se buscan los sinónimos de cada palabra que aparece en cada *query*, así se expande la búsqueda.
- El peso del sinónimo no es el mismo que el de la palabra original. Se divide el peso original entre el número de sinónimos que existen de esa palabra.

```
}
if(querySinonimosAux!=null && !querySinonimosAux.isEmpty()){
    double peso = ((double)1/querySinonimosAux.size());
    for (int i = 0 ; i<querySinonimosAux.size() ; i++){
        if(!querySinonimos.containsKey(querySinonimosAux.get(i))){
            querySinonimos.put(querySinonimosAux.get(i), peso);
        }
    }
}
```

MÓDULO I: EXPANSIÓN DE CONSULTA 2

- Los nombres propios se toman como entidades, pero también se tiene en cuenta el caso *LowerCase*.
- Por ejemplo, si aparece *Photoshop* en una consulta, la búsqueda se expande con la palabra *photoshop*, pero se le aplica la mitad del peso que tenía la palabra original.

```
List<String> filtered = querySinonimos.keySet()
        .stream()
        .filter(p -> !p.equals(p.toLowerCase()))
        .collect(Collectors.toList());

for(int i=0;i<filtered.size();i++){
    querySinonimos.put(filtered.get(i).toLowerCase(), querySinonimos.get(filtered.get(i))/2 );
}
```

MÓDULO II: RECONOCIMIENTO DE ENTIDADES

- Para gestionar las entidades, creamos un nueva colección y un índice en Mongo, como explicamos en la primera parte de la presentación.
- Ahí se guardan las entidades que reconocemos de los .**txt** dados y se almacena la entidad junto con los documentos en los que aparece.
- En el caso de existir alguno de estos nombres propios en la consultas, le daremos más peso a esa palabra en el documento.
- El peso que tiene la entidad en el documento se multiplica por 30.

RESULTADOS

Tiempo de indización de los documentos: Aproximadamente **8 minutos***.

- Creación del diccionario de palabras.
- Creación del diccionario de entidades.

Recursos utilizados:

- MacBook Pro 2015, 2,7 GHz Intel Core i5
- RAM: 8 GB
- Base de datos: MongoDB

**Sin los índices de Mongo, el tiempo obtenido fue de aprox. 29 horas*

MÉTRICAS DE EVALUACIÓN

Average Precision	
Q1	0,5709183673
Q2	0,7878306878
Q3	1
Q4	0,5932539683
Q5	0,325
Q6	0,9526289683
Q7	0,8301587302
Q8	0,9095238095
Q9	0,8875220459
Q10	0,8875220459
Q11	0,5416666667
Q12	0,9067956349
Q13	0,962654321
Q14	0,7928571429
Q15	0,6142857143
Q16	0,8806122449
Q17	0,5494047619
Q18	0,9317956349
Q19	0,6679138322
Q20	0,7180555556

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**	**Q8**	**Q9**	**Q10**	**Q11**	**Q12**	**Q13**	**Q14**	**Q15**	**Q16**	**Q17**	**Q18**	**Q19**	**Q20**
D1	0	1	2	1	0	1	2	1	2	1	0	2	2	1	1	2	0	2	0	1
D2	0	1	1	0	0	1	1	1	1	1	1	2	2	2	0	2	1	1	2	0
D3	1	1	1	0	0	1	1	1	1	1	1	1	1	0	0	1	0	1	2	1
D4	1	0	1	2	1	1	0	1	0	0	0	0	1	0	1	1	2	2	0	1
D5	1	0	2	0	2	1	0	1	2	1	1	1	1	1	0	0	0	0	1	0
D6	0	0	2	0	0	0	2	0	2	1	0	2	2	0	2	0	1	2	1	2
D7	1	1	1	1	0	1	1	0	1	1	0	1	0	1	1	2	2	1	1	0
D8	1	0	1	0	0	1	0	0	1	1	0	2	2	0	0	2	1	2	2	2
D9	2	1	1	1	0	2	0	1	1	1	0	1	2	0	0	0	0	2	2	0
D10	1	1	1	0	0	0	1	2	2	1	1	0	1	0	0	1	1	0	0	1

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Q15	Q16	Q17	Q18	Q19	Q20
p@5	0,60	0,60	1,00	0,40	0,40	1,00	0,60	1,00	0,80	0,60	0,60	0,80	1,00	0,60	0,40	0,80	0,40	0,80	0,60	0,60
p@10	0,70	0,60	1,00	0,40	0,20	0,80	0,60	0,70	0,90	0,90	0,40	0,80	0,90	0,40	0,50	0,70	0,60	0,80	0,70	0,60

POSIBLES MEJORAS

- Reconocimiento de entidades más eficiente interpretando *Who, Where, When,* etc… mediante el uso de herramientas como *FreeLing*.

- Uso de técnicas de stemming y lematización.

¿PREGUNTAS?