Bases de données documentaires et distribuées
Cours NFE04
RequÃłtes boolÃľennes

Auteurs : Raphaël Fournier-S'niehotta, Philippe Rigaux, Nicolas Travers
prénom.nom@cnam.fr

Département d'informatique
Conservatoire National des Arts & Métiers, Paris, France

Recherche de documents

- Possibilités très sophistiquées de Solr
- La manière d'effectuer une recherche varie en fonction de :
  - la syntaxe de la requête (de très structurée avec booléen sur des champs) à liste de mots-clefs
  - le **classement** du résultat
    - Solr renvoie les documents "pertinents" (pas seulement ceux qui correspondent exactement aux critères)
    - différence avec BDD (relationnelle)

Consulter l'index

- le paramètre qu'on utilise ici, c'est "q" (**query**)
- proposé par défaut dans l'interface comme "*.*" (**tous** les documents de l'index)
- **fq** : pour **filter query**, pour interroger non pas l'index entier mais un résultat pré-calculé et stocké en **cache**
- **sort**, pour trier le résultat
- **start** et **row**, les paramètres classiques de pagination du résultat
- **fl** pour **field list**, la liste des champs (stockés) à inclure dans le résultat
- **df**, le champ à interroger si non spécifié dans la requête
  (la valeur par défaut est indiqué dans la configuration et vaut en principe **text**, le champ dans lequel nous avons placé toutes nos chaînes de caractères) ;
- enfin, on trouve la liste des **query parsers** disponibles ;
  un **query parser** correspond à une syntaxe d'interrogation particulière.

Exemple d'utilisation des paramètres

- avec **sort=title asc** : tri du résultat sur le titre, en ordre ascendant ;
- avec **fl=title, year**, restriction des champs dans les documents du résultat ;
- avec **start=10, rows=9**, récupération des documents classés entre les positions 10 à 19 ;
- avec **q=Alien**, vous devriez retrouver le document Solr correspondant au film **Alien** ;
- avec **q=Alien** mais **df=summary**, vous ne devriez rien trouver ;
- avec **q=Vertigo**, **df=text**, vous devriez retrouver le film Vertigo ;

Paramètres

- le document résultat ne montre que ceux qui ont été définis dans le schéma comme “stockés”
- les autres champs sont utiles pour la recherche, mais on ne peut pas récupérer leur valeur
- En revanche, il est possible d'obtenir des informations **calculées** par Solr, sous forme de (pseudo-)champ :
  - ex. : le **score**, qui évalue la pertinence d'un document pour une recherche
  - essayer avec **title, year, score** dans **fl** (et une recherche par mot-clef (ex. : **fin**))
  - on vérifie que l'on a un classement par score
  - essayer en rajoutant **[explain style=nl]**

Les requêtes

- Solr fournit plusieurs interpréteurs de requêtes
- chacun reconnaît des syntaxes légèrement différentes
- l'interpréteur par défaut, "DisMax", est le plus intuitif
- mais pas toujours le plus précis

Termes

- Notion de base : le **terme**
- c'est un mot au sens usuel
- ou une séquence de mots entre apostrophes

Interroger l'index "collection1" avec :

```
hard drive
```

Puis :

```
"hard drive"
```

- Première recherche : documents avec "hard", "drive" ou les deux
- Deuxième : seulement "hard drive" (côte à côte)

Termes (suite)

- Dans Solr, la recherche d'un terme s'effectue toujours sur un champ.
- La syntaxe complète pour associer le champ et le terme est :

```
champ:terme
```

- si non précisé, c'est le champ par défaut qui est utilisé
- pratique courante : concaténer toutes les chaînes de caractères en un champ "text"général, défini par défaut
- Nos requêtes deviennent :

```
text:hard text:drive
```

- et

```
text:"hard drive"
```

Termes (suite)

- Les valeurs des termes (dans la requête) et le texte indexé sont tous deux soumis à des transformations spécifiées dans le schéma.
- Une transformation simple est de tout transcrire en minuscules.

```
text:"Hard Drive"
```

- Les transformations appliquées à la requête ET au texte indexé doivent être cohérentes : si les termes sont transformés en majuscules, et le texte indexé en minuscules, on n'aura jamais de résultat !

Termes (suite)

On peut spécifier des termes (pas des séquences) incomplets

- le ' ?' indique un caractère inconnu
  - "opti ?al" désigne "optimal", "optical", etc.
- le '*' indique n'importe quelle séquence de caractères
  - "opti*" pour toute chaîne commençant par "opti"

Approximations avec "∼" :

- Rechercher "optimal" et "optimal∼"
- 0 et 1 résultat ("optical")
- Proximité des termes par une distance d'édition :
  (nb opérations pour passer de "optimal" à "optical")

Intervalles :

- [] bornes comprises
- { } bornes exclues

```
%price:[100 TO 200]
```

Requêtes Booléennes

- Les critères peuvent être combinés avec des opérateurs Booléens : **AND**, **OR** et **NOT**
- Attention : majuscules

```
%price:[100 TO 300] OR popularity:5
%price:[100 TO 300] AND NOT popularity:5
%popularity:6 AND features:matrix
```

- Par défaut, c'est un **OR** qui est appliqué
- Recherche sur plusieurs critères ramène l'union des résultats sur chaque critère pris individuellement
- La requête suivante recherche les produits Dell **ou** dont la popularité est égale à 6 :

```
%popularity:6  manu:Dell
```

Opérateur +, classement

- préfixe d'un nom de champ, il indique que la valeur du champ **doit** être égale au terme
- il existe également un opérateur −, équivalent au **NOT**
- recherche des documents dont la popularité est 6 (obligatoire) et qui peuvent être produits par Dell ou un autre constructeur

```
%+popularity:6  manu:Dell
```

- différence avec ce qui précède : le **classement** du moteur
- illustre la différence entre recherche d'information et interrogation de bases de données

Interprêter un classement est parfois délicat :

```
%+popularity:6  cat:electronics

%+popularity:6  -manu:Dell
```