

Cours elasticsearch

Johnny Mariéthoz

RERO

Cours HEG

Table des matières I

1 Moteurs de recherche : la partie visible

2 Théorie : Information Retrieval

- Deux étapes
- Principes
- Mesures
- Limitations

Table des matières II

3 Elasticsearch

- Concepts
- "Analyzer" et "Mapping"
- Query
- Facettes et filtres
- Query_string
- Autres

4 En pratique

Moteurs de recherche

1 Moteurs de recherche : la partie visible

Interface utilisateur

- Champ de recherche entrée du texte de recherche
- Facettes fréquence d'un terme
- Filtres réduit la liste des résultats (click sur les facettes)
- Tris ordonnancement des résultats
 - par titre
 - par auteur
 - par pertinence
 - par date de publication
- Scoring pondération des champs pour le tris par pertinence
- Pagination nombre de résultats par page

Théorie : Information Retrieval

2 Théorie : Information Retrieval

- Deux étapes
- Principes
- Mesures
- Limitations

Définition

*La recherche d'information (RI) est le domaine qui étudie la manière de **retrouver des informations** dans un **corpus**. [Wikipédia]*

Indexation

- création d'un **index de descripteurs**
- lors de l'ajout d'un document
- peut **prendre du temps** (machine)

Indexation

- création d'un **index de descripteurs**
- lors de l'ajout d'un document
- peut **prendre du temps** (machine)

Recherche

- comparaison d'une **requête** avec un lot de **documents**
- doit être **rapide**

Comment ça marche ?

Produit interne (Inner product) :

$$\begin{array}{c} \text{term}_1 \\ \text{term}_2 \\ \vdots \\ \text{term}_n \end{array} \begin{array}{c} \mathbf{q} \\ \left(\begin{array}{c} 1 \\ 0 \\ \vdots \\ 0 \end{array} \right) \end{array} \cdot \begin{array}{c} \mathbf{d}_1 \quad \mathbf{d}_2 \quad \dots \quad \mathbf{d}_n \\ \left(\begin{array}{cccc} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{array} \right) \end{array}$$

Note : dans la vraie vie les vecteurs sont sparse et on ajoute des poids : http://en.wikipedia.org/wiki/Vector_space_model

Principe de génération des descripteurs (termes)

- pré-traitements
 - **stemming** : racinisation ou désuffixation
 - **tokenization** : décomposition en mots
- extraction des descripteurs sous forme de vecteur d'entiers de la taille du nombre de descripteurs
- rajout de règles empiriques, recherche n-gram, synonymes, etc.
- un temps de réponse faible => optimisation + vecteur sparse (Lucene)

Mesures de performance

- "precision and recall"
 - on fixe un nombre de résultat (par ex. 10) et on **compte** le nombre de résultats pertinents
- demande des données **annotées**

Limitations

- des avions : stop word
- pain (boulangerie), pain(english)
- thé - the
- **qualité** des données

Elasticsearch

3 Elasticsearch

- Concepts
- "Analyzer" et "Mapping"
- Query
- Facettes et filtres
- Query_string
- Autres

Concepts similaires à SQL

index base de donnée

document type table

document ligne d'une table

term colonne

Configuration de l'index (avant indexation)

```
{  
  "analyzer" : {  
    "custom_analyzer" : {  
      "tokenizer" : "keyword",  
      "filter" : ["standard", "asciifolding",  
                  "lowercase"]  
    }  
  }  
}
```


Mapping (avant indexation)

Ajout de connaissances "a priori"

```
"language" : {  
  "type": "string",  
  "index": "not_analyzed"  
}
```

- type de données (integer, string, etc.)
- "index" : "not_analyzed" == "analyzer" : "keyword"

Mapping complexe

```
"authors" : {  
  "properties" : {  
    "other_authors": {  
      "type": "string",  
      "fields" : {  
        "other_authors": {  
          "copy_to" : ["facet_authors"],  
          "type": "string",  
          "analyzer": "simple"  
        }  
      }  
    }  
  },  
  "facet_authors" : {  
    "type" : "string",  
    "index": "not_analyzed"  
  }  
}
```

```
{
  "size": 20,
  "from": 0,
  "query": {
    "filtered": {
      "query": {},
      "filter": {}
    }
  },
  "aggs": {},
  "highlight": {},
  "sort": []
}
```

Facets / aggregators (lors de la recherche)

```
"aggs": {  
  "language": {  
    "terms": {  
      "field": "language",  
      "size": 10,  
      "order": {  
        "_count": "desc"  
      }  
    }  
  }  
}
```

- auteurs, type de documents, etc.
- approximation du nombre d'occurrences

Filtres (recherche)

- comme une requête **booléenne**
- plus **rapide** que "query"
- utilisé par les facettes

Requête utilisateur (texte de recherche)

- Lucene query (support de title :)
- boosting (pondération des termes, par ex. un titre a plus de poids qu'une note)

```
"query_string": {  
  "fields" : ["title.title^4",  
              "title.subtitle^2",  
              "abstract"],  
  "query": user_query  
}
```

"Highlights" (recherche)

- mise en évidence du descripteur correspondant à la recherche
- supporte le "stemming"

Tris (recherche)

- tris par terme ou par **pertinence** (valeur par défaut)

```
"sort" : [  
  {  
    "title.title" : {  
      "order" : "asc"  
    }  
  }  
]
```


En pratique

4 En pratique

Outil : "es_lab"

- utilisation de l'extension "sense"
- basé sur python-Flask
- conversion des données MarcXML en JSON
- script d'indexation
- configuration des indexes et des termes
 - marc2json
 - facettes
 - highlight
 - mapping
- interface utilisateur (page web)