Data MGnt

26/10/2015

1 Understanding the Lucene API

1.1 Does the command line use demo use stopword removal ?

Oui, l'exemple utilisé dans la démo utilise le *stopword removal*. On le remarque en essayant de rechercher les deux chaînes suivantes :

- "string"
- "the string"

Pour les deux, nous obtenons 1699 documents correspondants. Il est donc évident que le mot *the*, qui est un stopword, n'a pas été utilisé pour la requète. Dans le cas contraire, nous aurions eu l'ensemble des documents retournés (plus de 1300).

De plus, la chaîne "*the*" ne retourne aucun document.

1.2 Does the command line use demo use stemming ?

Non, l'exemple utilisé dans la démo n'utilise pas de méthode de *stemming*. Pour le voir, nous avons testé plusieurs couples de chaînes :

- *example* et *exampl* : Le premier renvoi 291 documents tandis que le second aucun
- *man* et *men* : Le premier nous donne 3 résultats tandis que le second aucun
- *reason* et *reasons* : Le premier renvoi 21 résultats tandis que le second en renvoi 22. Chacun de ces mots est considéré de manière différente. Ainsi, on peut voir que l'analyseurs de fait pas stemming.

1.3 Is the search of the command line demo case insensitive ?

Oui, l'exemple utilisé dans la démo n'est pas sensible à la casse. En cherchant les mots *man*, *Man*, *maN*, *mAn* et *MAN* nous trouvons exactement les mêmes résultats

1.4 Does it matter whether stemming occurs before or after stopword removal?

Cela dépend de la liste de *stop words*. Si la liste de stop word a des *stems*, alors il est mieux d'appliquer le *stop word removal* après *stemming*. Sinon, il est préférable de l'appliquer avant *stemming*.

Mais cela dépend aussi de la langue utilisée, dans certaines langues il est plus difficile de créer une liste de stop word sans *stem*.

2 Indexing and Searching the CACM collection with Lucene

2.1 Explain which field type can be used for title, summary and author

- **Title**: TextField car le titre doit être *tokenized*.
- **Summary**: TextField, car le résumé doit être *tokenized*.
- **Author**: String Field pour chacun des auteurs car chaque auteur représente une donnée atomique

2.2 What should be added to the code to have access to the term vector in the index?

```
FieldType fieldType = new FieldType();
fieldType.setIndexOptions(IndexOptions.DOCS);
fieldType.setTokenized(true);
fieldType.setStored(true);
fieldType.setStoreTermVectors(true);
fieldType.freeze();
Field fieldSum = new Field("summary", fields[3], fieldType);
doc.add(fieldSum);
```

3 Using different Analyzers

L'image arrive :-)