

# NFE204

Recherche d'information : l'indexation  
S1 : l'analyse de documents

Auteurs : Raphaël Fournier-S'niehotta, Philippe Rigaux  
(fournier@cnam.fr, philippe.rigaux@cnam.fr)

EPN Informatique  
Conservatoire National des Arts & Métiers, Paris, France

# Plan du cours

## 1 Où en est-on ?

## Lors du cours précédent

### Introduction générale à la RI

- Contexte d'utilisation
- Évaluation des moteurs
- Index inversés : structure de données performante
- Présentation d'ElasticSearch

### Aujourd'hui

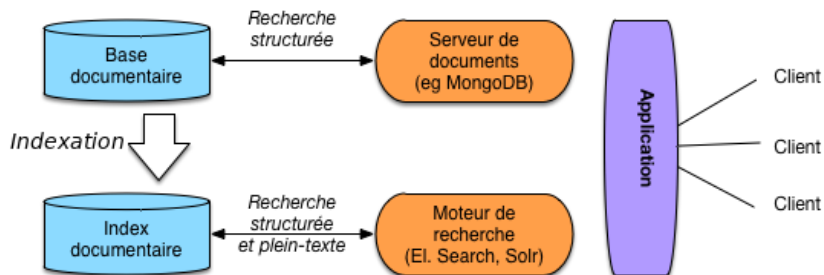
- Comment transformer les données pour des recherches efficaces ?
- Utilisation d'un schéma
- Détail de l'analyse du texte
- Reprise de la mise en pratique d'ElasticSearch
- Interactions avec l'API REST
- Premiers documents, premières requêtes

# Plan du cours

## 2 Indexation

- Tokenisation et normalisation
- Racinisation
- Mots vides, synonymes, cas particuliers...

## Bases documentaires et moteur de recherche



## Pré-traitement

- Dès qu'un document est un peu complexe, on ne peut pas le découper arbitrairement sans appliquer un pré-traitement réfléchi
- sinon :
  - que devient "pomme de terre" ?
  - on stocke des listes très longues pour certains mots...Retour à ES
  - l'utilisateur doit faire très attention ("loup", "Loup", "loups" sont différents)
- il faut procéder à une **analyse du contenu**

## Rôle de l'analyse

L'analyse des textes permet d'effectuer une forme de normalisation / unification pour être moins dépendant de la forme du texte.

- un document parle de loup même si on y trouve les formes "loups", "Loup", "louve", etc.
- un document parle de travail quelle que soit la forme du verbe "travailler" ou de ses variantes.
- Jusqu'où va-t-on? Traductions (loup = wolf = lupus) ? Synonymes (loup = prédateur) ? (sujet de recherches)

Pour moins dépendre de la forme on applique des **transformations**.

## Impact de l'analyse

Bien comprendre :

Plus on normalise, plus on diminue la précision.

Car des mots distincts sont unifiés (cote, côte, côté, etc.)

Plus on normalise, plus on améliore le rappel.

Car on met en correspondance les variantes d'un même mot, d'une même signification (conjugaisons d'un verbe).



## Impact de l'analyse

### Très important

La même transformation doit être appliquée aux documents **et** à la requête.

Pourquoi ?

- sinon, il se crée un décalage entre ce qui a été analysé et ce qui peut être réellement recherché

## Les phases de l'analyse

Important: identification de quelques méta-données (la langue), prise en compte du contexte (quels documents pour quelle application).

Puis, de manière générale :

- **Tokenization** : découpage du texte en "mots"
- **Normalisation** : majuscules ? acronymes ? apostrophes ? accents ?  
Exemple : *Windows* et *window*, U.S.A vs USA, *l'étudiant* vs *les étudiants*.
- **Stemming** ("racinisation"), **lemmatization**  
Prendre la racine des mots pour éviter le biais des variations (étudier, étudiant, étude, etc.)
- **Stop words**, quels mots garder ?  
Mots très courants peu informatifs (le, un à, de).

C'est de l'art et du réglage... Dans ce qui suit : introduction / sensibilisation aux problèmes.

## Identification de la Langue

Comment trouver la langue d'un document ?

- Méta information (dans l'entête HTTP p.e.): pas fiable du tout.
- Par le jeu de caractères, pas assez courant!

한글  
カタカナ  
مَرْكَب  
Gharbi  
pom

## Identification de la Langue

Comment trouver la langue d'un document ?

- Méta information (dans l'entête HTTP p.e.): pas fiable du tout.
- Par le jeu de caractères, pas assez courant!

한글  
カタカナ  
ދިވެހި  
Għarbi  
þom

**Respectivement:** Coréen, Japonais, Maldives, Malte, Islandais.

- Par extension: séquences de caractères fréquents, ( $n$ -grams)
- Par techniques d'apprentissage (classifiers)

Des librairies font ça très bien (e.g., Tika, <http://tika.apache.org>)

# Tokenisation

## Principe

Séparation du texte en **tokens** (“mots”)

Pas du tout aussi facile qu'on le dirait !

- Dans certaines langues (Chinois, Japonais), les mots **ne sont pas** séparés par des espaces.
- Certaines langues s'écrivent de droite à gauche, de haut en bas.
- Que faire (et de manière **cohérente**) des acronymes, élisions, nombres, unités, URL, email, etc.

# Tokenisation

- **Mots composés** : les séparer en *tokens* ou les regrouper en un seul ?
  - 1 Anglais : *hostname*, *host-name* et *host name*, ...
  - 2 Français : Le Mans, aujourd'hui, pomme de terre, ...
  - 3 Allemand : *Lebensversicherungsgesellschaftsangestellter* (employé d'une société d'assurance vie)

Que faire si l'utilisateur cherche *hostname* et qu'on a normalisé en *host-name* ?

Majuscules, ponctuation ? Une solution simple est de normaliser (minuscules, pas de ponctuation).

## Exemple pour notre petit jeu de données

On met en minuscules, on retire la ponctuation.

- $d_1$  le loup est dans la bergerie
- $d_2$  le loup et les trois petits cochons
- $d_3$  les moutons sont dans la bergerie
- $d_4$  spider cochon spider cochon il peut marcher au plafond
- $d_5$  un loup a mangé un mouton les autres loups sont restés dans la bergerie
- $d_6$  il y a trois moutons dans le pré et un mouton dans la gueule du loup
- $d_7$  le cochon est à 12 euros le kilo le mouton à 10 euros le kilo
- $d_8$  les trois petits loups et le grand méchant cochon

On considère que l'espace est le séparateur de tokens.

## Stemming (racine), lemmatization

### Principe

**Confondre** toutes les formes d'un même mot, ou de mots apparentés, en une seule **racine**.

Stemming Morphologique. Retire les pluriels, marque de genre, conjugaisons, modes, etc.

- Très dépendant de la langue: *geese* pluriel de *goose*, *mice* de *mouse*
- Difficile à séparer d'une analyse linguistique ("Les poules du couvent couvent", "la petite brise la glace": où est le verbe?)

Stemming lexical Fondre les termes proches lexicalement: "politique, politicien, police (?)” ou "université, universel, univers (?)”

Stemming phonétique. Correction fautes de frappes, fautes orthographes



## Exemple de stemming

On retire les pluriels, on met le verbe à l'infinitif.

- d*<sub>1</sub> le loup etre dans la bergerie
- d*<sub>2</sub> le loup et les trois petit cochon
- d*<sub>3</sub> les moutons etre dans la bergerie
- d*<sub>4</sub> spider cochon spider cochon il pouvoir marcher au plafond
- d*<sub>5</sub> un loup avoir manger un mouton les autres loups etre rester dans la bergerie
- d*<sub>6</sub> il y avoir trois mouton dans le pre et un mouton dans la gueule du loup
- d*<sub>7</sub> le cochon etre a 12 euro le kilo le mouton a 10 euro le kilo
- d*<sub>8</sub> les trois petit loup et le grand mechant cochon

## Suppression des *Stop Words*

### Principe

On retire les mots porteurs d'une information faible afin de limiter le stockage.

articles: *le, le, ce*, etc.

verbes "fonctionnels" *être, avoir, faire*, etc.

conjunctions: *that, and*, etc.

etc.

- ♣ Maintenant moins utilisé car (i) espace de stockage peu coûteux et (ii) pose d'autres problèmes ("pomme de terre", "Let it be", "Stade de France")

## Autres problèmes, en vrac

### Majuscules / minuscules

Lyonnaise des Eaux, Société Générale, etc.

### Acronymes

CAT = *cat* ou *Caterpillar Inc.*? M.A.A.F ou MAAF ou Mutuelle ... ?

### Dates, chiffres

Monday 24, August, 1572 – 24/08/1572 – 24 août 1572 10000 ou 10,000.00 ou 10,000.00

### Accents, ponctuation

résumé ou résumé ou resume...

- ♣ Dans tous les cas, les même règles de transformation s'appliquent aux documents ET à la requête.

## Exemple avec suppression des *stop words*

Voici une solution possible.

- $d_1$  loup etre bergerie
- $d_2$  loup trois petit cochon
- $d_3$  mouton etre bergerie
- $d_4$  spider cochon spider cochon pouvoir marcher plafond
- $d_5$  loup avoir manger mouton autres loups etre rester bergerie
- $d_6$  avoir trois mouton pre mouton gueule loup
- $d_7$  cochon etre 12 euro kilo mouton 10 euro kilo
- $d_8$  trois petit loup grand mechant cochon

On a gardé les verbes fonctionnels (être, avoir).