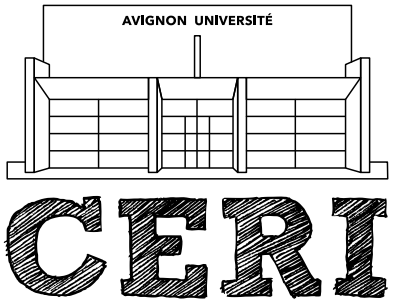

Centre d'Enseignement et de Recherche en Informatique

Partie 3
Construction du lexique

Vincent Labatut

Laboratoire Informatique d'Avignon – LIA EA 4128

vincent.labatut@univ-avignon.fr

2019/20

M2 ILSEN

UE Ingénierie du document et de l'information

UCE3 Indexation & Recherche d'information

AVIGNON UNIVERSITÉ

Plan de la séance

1. Constitution de la collection
2. Définitions préliminaires
3. Étape de tokénisation
4. Mots vides
5. Étape de normalisation

Section 1

Constitution de la collection

AVIGNON
UNIVERSITÉ

Constitution de la collection
Récupération et accès aux données

- Acquisition des données
  - Tâche complètement dépendante de la source : Web, BD, fichiers locaux...
  - Peut nécessiter un prétraitement spécifique : crawl, requêtage, lecture...
- Nature des fichiers
  - **alert** : texte, binaire (ex. MS Word, ZIP)
  - **alert** : ASCII, Unicode, propriétaire
  - **alert** d'entités (ex. SGML : `&amp;`)
  - **alert** de la partie textuelle (ex. XML, PS, PDF)

  → peut être très difficile à automatiser (ex. pages Web)
- Identification de la langue
  - Unilingue vs. multilingue

Constitution de la collection
Unité de document

- Que choisir comme une unité de document ?
- 1 document $\stackrel{?}{=}$ n fichiers
  - Ex. : `latex2html` transforme un document LaTeX en un site web contenant plusieurs pages
- n documents $\stackrel{?}{=}$ 1 fichier
  - Ex. : un fichier `mbox` peut contenir une séquence d'emails
  - Ex. : un fichier ZIP peut contenir une collection de documents
- Niveau de granularité
  - Ex. : livres vs. chapitres
  - Trop grande $\rightarrow$ faible précision
  - Trop petite $\rightarrow$ faible rappel

Section 2

Définitions préliminaires

AVIGNON
UNIVERSITÉ

Définitions préliminaires
Tokens & tokénisation

Segmentation (ou Tokénisation)

Découpage du texte constituant les documents du corpus, de manière a produire des séquences de **segments** (ou **tokens**) correspondant approximativement à des **mots** ou **expressions**.

Ex. : Friends, Romans, Countrymen, lend me your ears ;
→ Friends Romans Countrymen lend me your ears

Token

Chaîne de caractères apparaissant dans un document du corpus, et considérée comme une unité sémantique. Concrètement, un token correspond à un **mot** ou à un **groupe de mots**. Un token est une **instance** de **type**.

Ex. : le mot Friends apparaissant dans la phrase d'exemple

Définitions préliminaires
Types & Termes

Type

Classe de tous les **tokens** correspondant à la même chaîne de caractères. Un type apparaît généralement **plusieurs fois** dans le corpus, sous la forme de plusieurs tokens **distincts**.

Ex. : le mot Friends *en général* (par opposition à une instance particulière)

Terme

Version **normalisée** d'un **type**, qui est utilisée dans le **lexique** d'un index. Il ne s'agit pas forcément d'un mot.

Ex.1 : le mot Friends *en général* (pas une instance particulière)
Ex.2 : concept ontologique (=code associé à une sémantique donnée)

Définitions préliminaires
Tokens vs. types vs. termes

- Comparaison 1 :
  - Phrase : The friends of my friends are my friends
  - 8 tokens : The, friends, of, my, friends, are, my, friends
  - 5 types : The, friends, of, my, are
  - Probablement 1 seul terme : friend
- Comparaison 2 :
  - Phrase : Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo [Wik19]
  - 8 tokens : Buffalo, buffalo, Buffalo, buffalo, buffalo, buffalo, Buffalo, buffalo
  - 2 types : Buffalo, buffalo
  - Probablement 1 seul terme : buffalo

Section 3

Étape de tokénisation

AVIGNON
UNIVERSITÉ

Étape de tokénisation
Approche standard

- La même segmentation doit être effectuée sur la requête
- Comment segmenter ?
  - Approche *standard* : considérer les caractères non-alphanumériques comme des séparateurs
  - Ex. : Je suis là $\rightarrow$ 3 tokens
- Principaux problèmes rencontrés :
  - Caractères d'espacement
    - Ex. : Le laboratoire est ouvert $\rightarrow$ 4 tokens
    - Mais : plate forme $\rightarrow$ 2 tokens ?
  - Ponctuation
    - L'homme n'avait jamais pris ce chemin $\rightarrow$ 8 tokens
    - Mais : aujourd'hui $\rightarrow$ 2 tokens ? U.S.A. $\rightarrow$ 3 tokens ?
  - Les deux :
    - J'ai pris le vol Los Angeles-Paris $\rightarrow$ 8 tokens ?

Étape de tokénisation
Autres problèmes

- Autres problèmes :
  - Valeurs numériques
    - Numéros de téléphone 04 90 12 34 56
    - Dates 10/09/2014
  - Codes et noms propres
    - Le langage C++
    - Le bombardier B-52
  - Adresses
    - email : `bozo@mysite.org`
    - URL : `http://www.univ-avignon.fr`
    - IP : 172.16.254.1

Étape de tokénisation
Problèmes relatifs à la langue

- Problème de segmentation de mots :
  - Mots construits (allemand, turc...)
    - Computerlinguistik (2 tokens) pour : *linguistique computationnelle*
    - Çekoslovakyalılaştıramadıklarımızdanmısınız ? pour : *Êtes-vous l'une de ces personnes que nous n'avons pas pu tchécoslovaquiser ?*
  - Aucun espace (chinois, japonais...)
  - Solutions :
    - Prétraitement linguistique de segmentation des mots : manuel, automatique (ML)
    - Utilisation de n-grammes de caractères au lieu de mots
- Texte bidirectionnel : contient des parties respectant différents sens de lecture
  - Ex. : texte arabe avec valeurs numériques en chiffres indo-arabes (exemple ici)

Section 4

Mots vides

AVIGNON
UNIVERSITÉ

Mots vides
Notion de mot vide

Mot vide (EN : stop word)

Mot si **fréquent** que son sens n'est **pas** jugé **pertinent**.

Ex. : le, la, un, cette...
...mais la notion de pertinence est relative au contexte du corpus

On ignore les mots vides dans le lexique

- Intérêt : réduire significativement le nombre de postings dans l'index
- Perte d'information relativement minime → qualité des résultats à peu près équivalente

Mots vides
Fréquences d'un terme

Fréquence d'un terme

La fréquence d'un terme t dans un document d correspond à son **nombre** d'**occurrences** dans ce **document**, noté $tf(t, d)$.

Fréquence de collection d'un terme

La fréquence de collection d'un terme t est le **nombre** total d'**occurrences** de ce terme dans le **corpus**, noté $cf(t)$.
Formellement :

$$cf(t) = \sum_{d \in \mathcal{C}} tf(t, d).$$

Rappel : la fréquence de document $df(t)$ est le nombre de documents de $\mathcal{C}$ contenant t

Mots vides
Méthodes de détection

- Deux méthodes complémentaires pour détecter les mots vides :
  1. Filtrage des termes les plus fréquents
     - On utilise la fréquence de collection $cf(t)$ du terme t
     - On considère comme mots-vides tous les termes dépassant une certaine fréquence
     - Ce seuil peut être fixé arbitrairement ou empiriquement (en considérant les données)
  2. Utilisation de listes prédéfinies
     - Ex. : articles (la, le, les...), pronoms (je, tu, elle...), conjonctions (qui, que, dont...), etc.
- Contre-exemples :
  - Le vol pour Londres est plus précis que : vol, Londres
  - To be or not to be $\rightarrow$ uniquement des mots vides

Section 5

Étape de normalisation

AVIGNON
UNIVERSITÉ

Étape de normalisation
Notion de normalisation

Normalisation

Action d'associer un **terme unique** à une **classe** de types jugés équivalents, puis de **substituer** ce terme aux **tokens** instanciant l'un de ces types dans le texte.

Exemples :

- Le terme chien représentant les types chien, chiens, chienne, et chiennes
- Le terme plateforme représentant les types plate-forme, plate forme et plateforme
- Le terme chant représentant chanter, chante, chantes, chantons, chantez, chantent...
- Le terme AB1267Q représentant voiture, autombile, caisse, bagnole

Étape de normalisation
Méthodes générales

- Méthode 1 : classes d'équivalence
  - Créer (manuellement ou automatiquement) des maps spécifiques token → terme
  - Ou : utiliser des règles générales de transformation (ex. supprimer les tirets)
- Méthode 2 : lier les types
  - Indexer tous les types
  - Utiliser des maps *type* → *type plus spécifique* et les appliquer aux requêtes
  - Ex. : indexer window, windows et Windows
    - Windows → Windows
    - windows → Windows, windows, window
    - window → window, windows
- Méthode 2 plus flexible, mais beaucoup plus coûteuse (temps et mémoire)

Étape de normalisation
Problèmes typographiques

Signe diacritique

Signe modifiant une lettre de l'alphabet : accent, cédille, tréma, etc.

- Signes diacritiques :
  - Approche standard : supprimer tous les signes
  - Problème : boite vs. boîte, jeûne vs. jeune
- Majuscules vs. minuscules :
  - Approche standard : tout convertir en minuscules
  - Mieux : seulement les mots en début de phrase
  - Problème : USA vs. usa
- Compromis entre :
  - Perte d'information
  - Performance obtenue
  - Utilisation effective
    - Ex. : accents omis dans la requête pour des raisons de rapidité, paresse, habitude, contraintes...

Étape de normalisation
Problèmes linguistiques

- Problèmes dépendant de la langue/système d'écriture :
  - Multiples formes :
    - Français : l' = le, la
  - Variantes orthographiques
    - Anglais UK vs. US : colour vs. color
    - Translitérations Chebyshev vs. Tchebycheff
  - Conventions orthographiques
    - Allemand : ü=ue (Schütze = Schuetze)
    - Suédois : å=aa (Ålborg = Aalborg)
  - Systèmes d'écriture multiples, ex. Japonais :
    - Kanji 漢字 : logogrammes
    - (Ateji : séquence de kanjis à valeur phonétique)
    - Kana : syllabaires (hiragana ひらがな , katagana カタカナ)
    - Rōmaji : alphabet latin

Étape de normalisation
Problèmes de flexion

Flexion

Modification du mot permettant de représenter certains **traits grammaticaux** (genre, nombre, cas, temps, voix, classe lexicale...).

- Problèmes liés aux flexions :
  - Un même mot peut subir plusieurs flexions simultanément
    - Ex.1 : l'université vs. les universités
    - Ex.2 : il écrivit vs. nous écrirons
  - Des mots dérivés de la même racine peuvent être jugés équivalents
    - Ex. : démocratie vs. démocratisation
- Solution : se ramener à une forme neutre dans le lexique
  - Ex. : Nous écrirons à l'université → nous écrire université
  - Deux méthodes : racinisation vs. lemmatisation

Étape de normalisation
Opération de racinisation

Racinisation ou désuffixation (eng : Stemming)

Transformation d'un mot consistant à faire **disparaître** ses flexions en lui substituant sa **racine**.

- Ex. : $\text{frontal} \rightarrow \text{front}$; $\text{chercher} \rightarrow \text{cherch}$
- Réalisation : méthodes à base de règles de ré-écriture
  - Approche par dictionnaire : liste prédéfinie de paires (type, racine)
  - Approche algorithmique : Porter stemmer (EN) ; Carry (FR)
- Limites :
  - La racine ne correspond pas forcément à un mot réel
  - Confond les flexions (ex. $\text{saut}\underline{\text{er}}$ vs. $\text{saut}\underline{\text{ent}}$) et les dérivations ($\text{saut}\underline{\text{er}}$ vs. $\text{saut}\underline{\text{iller}}$)

Étape de normalisation
Opération de lemmatisation

Lemmatisation

Transformation d'un mot consistant à le ramener à une **flexion canonique** et lui substituant son **lemme**, i.e. le même mot sous une forme neutre (singulier-masculin, infinitif...).

- Ex. : $\text{écrirons} \rightarrow \text{écrire}$; $\text{universités} \rightarrow \text{université}$
- Réalisation : méthodes à base de règles (comme racinisation)
  - Approche par dictionnaire : liste prédéfinie de paires (type,lemme)
  - Approche algorithmique : règles définies manuellement ou apprises automatiquement
- Propriétés :
  - Le lemme correspond par définition à un mot réel
  - Ne tient compte que des flexions, pas des dérivations

Section 6

Conclusion

AVIGNON
UNIVERSITÉ

Concepts abordés dans cette partie

- Tokenisation
- Signe diacritique
- Normalisation
- Racinisation
- Lemmatisation
- Fréquence de collection
- Token
- Type
- Terme
- Mot-vide
- Lemme
- Flexion

Lectures recommandées

[MRS08] *Introduction to Information Retrieval*, chapitre 2.

[BCC10] *Information Retrieval : Implementing and Evaluating Search Engines*, chapitre 2.

[BR11] *Modern Information Retrieval : The Concepts and Technology behind Search*, chapitre 6.

[AG13] *Recherche d'information - Applications, modèles et algorithmes*, chapitre 2.

[CMS15] *Search Engines : Information Retrieval in Practice*, chapitres 4 & 5.

Références bibliographiques I

[AG13] M.-R. Amini et É. Gaussier. *Recherche d'information - Applications, modèles et algorithmes.* Paris, FR : Eyrolles, 2013. url : https://www.eyrolles.com/Informatique/Livre/recherche-d-information-9782212673760/.

[BR11] R. Baeza-Yates et B. Ribeiro-Neto. *Modern Information Retrieval : The Concepts and Technology behind Search.* 2nd Edition. Boston, USA : Addison Wesley Longman, 2011. url : http://people.ischool.berkeley.edu/~hearst/irbook/.

[BCC10] S. Büttcher, C. L. A. Clarke et G. V. Cormack. *Information Retrieval : Implementing and Evaluating Search Engines.* Cambridge, USA : MIT Press, 2010. url : http://www.ir.uwaterloo.ca/book/.

[CMS15] W. B. Croft, D. Metzler et T. Strohman. *Search Engines : Information Retrieval in Practice.* Pearson, 2015. url : http://www.search-engines-book.com/.

Références bibliographiques II

[MRS08] C. D. Manning, P. Raghavan et H. Schütze. *Introduction to Information Retrieval*. New York, USA : Cambridge University Press, 2008. url : http://www-nlp.stanford.edu/IR-book/.

[Wik19] Wikipedia. *Buffalo buffalo Buffalo buffalo buffalo buffalo Buffalo buffalo*. 2019. url : https://en.wikipedia.org/wiki/Buffalo_buffalo_Buffalo_buffalo_buffalo_buffalo_Buffalo_buffalo.