RECHERCHE D'INFORMATION & TRAITEMENT AUTOMATIQUE DU LANGAGE

Cours 1 : RI - introduction et indexation

6 mars 2023

Benjamin Piwowarski / Laure Soulier

ISIR INSTITUT DES SYSTÈMES INTELLIGENTS ET DE ROBOTIQUE

MLIA Machine Learning & Deep Learning for Information Access

RI - Introduction

La recherche d'information

Au centre du monde digital...

- Moteurs de recherche
- Recherche de documents
- Assistants vocaux
- ...

La recherche d'information

Au centre du monde digital...

- Moteurs de recherche
- Recherche de documents
- Assistants vocaux
- ...

... et en évolution

- Recherche Interactive
- Apprentissage Machine

Recherche d'Information

Définition

"Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."
C. Manning

- Application la plus courante : Moteurs de recherche

- Mais aussi dans : les entreprises, les bibliothèques numériques, domaines d'application (médecine, droit, ...), nos ordinateurs...

La donnée, l'or noir de l'information

SCIENCES SORBONNE UNIVERSITÉ

Données

- En 2020, 1.7MB généré chaque seconde (par personne)
- Facebook : 31.25 million de messages (par minute)

Informations

- 8 milliard de questions sur Google (chaque jour, soit 100 000 par seconde)
- 16% to 20% des questions sont *nouvelles*.

$\rightarrow$ RI : Collecter, organiser et identifier la bonne donnée au bon moment pour le bon utilisateur pour lui donner de l'information

Connaissances

Data Mining, Machine Learning

Des enjeux...

- Liés aux sources d'information
  - Texte : articles, livres (pdf, ps, ebook, html, xml, ...)
  - Images, Vidéos, Son, Musique
  - Pages/sites Web dynamiques
  - Médias sociaux (blogs, Twitter, ...) : dynamicité, structure relationnelle
  - Messageries - fils de discussion
  - Information majoritairement peu structurée, mais structures exploitables (HTML, XML), relations (web réseaux sociaux), hiérarchies, ...

Des enjeux...

- Liés à la diversité des demandes d'accès à l'information
  - Consultation (browsing)
  - questions booléennes, mots-clés
  - Recherche automatique (ex. robots)
  - Suivi d'évènements, analyse de flux
  - Extraction d'information du texte
  - ...

... Aux systèmes de RI

Problèmes de base pour construire un système d'accès à l'information

- Acquisition
  ☞ *crawling* et pré-traitement (diversité des types de documents)
- Représentation - indexation
  ☞ non structuré (texte, image), semi structuré (ex. vidéo, tableaux)
- Modèle de recherche
  ☞ présenter des informations pertinentes à l'utilisateur, ex. liste ordonnée selon un critère
- Interaction utilisateur
  ☞ feedback, recherche interactive, la RI est un processus centré utilisateur
- Evaluation
  ☞ Protocole d'évaluation, ex. Cranfield, mesures ex. rappel-précision
- Et puis pour les données du web
  ☞ Dynamicité, Performance, Passage à l'échelle (quantité de données (tera), stockage distribué), ...

Exemples de tâches en RI classique

- RI ad-hoc : Trouver parmi un ensemble d'articles ceux qui concernent un sujet spécifique : pertinence d'un document ?
- Faire un résumé du contenu d'un document ou d'un ensemble de documents (éventuellement sur un sujet)
- Structuration (classification) automatique d'un ensemble de documents (groupes)
- Trouver dans un document les passages pertinents, les informations pertinentes concernant un sujet (mots - phrases)
- Suivre dans une collection d'articles l'évolution d'un sujet, changements de sujets
- Guetter l'arrivée d'informations (appels d'offre, CFP, nouveaux produits, ...)
- Dialoguer avec les clients (ex. Hot Line, réclamations, ...) ....

Campagnes d'évaluation

Importance des campagnes

Les campagnes d'évaluation sont centrales pour le développement de la recherche d'information

ingrédients de base = documents, questions, jugements et métriques

Campagnes d'évaluation

Importance des campagnes

Les campagnes d'évaluation sont centrales pour le développement de la recherche d'information
ingrédients de base = documents, questions, jugements et métriques

Text REtrieval Conference (TREC)

Financé par NIST (département du commerce US)
`https://trec.nist.gov`
Tâches (en 2023) :

- Deep Learning Track
- Interactive Knowledge Assistance Track (iKAT)
- NeuCLIR Track (cross-language IR)
- Clinical Trials Track
- Tip-of-the-Tongue Track

Campagnes d'évaluation

Conference and Labs of the Evaluation Forum (CLEF)

Started from PROMISE Network of Excellence

`https://www.clef-initiative.eu`

- BioASQ - Large-scale biomedical semantic indexing and question answering
- CheckThat ! - Check-Worthiness, Subjectivity, Political Bias, Factuality, and Authority of News Articles and Their Sources
- DocILE - Document Information Localization and Extraction
- eRisk - Early Risk Prediction on the Internet
- EXIST - sEXism Identification in Social neTworks
- iDPP@CLEF - Intelligent Disease Progression Prediction
- ImageCLEF - Multimedia Retrieval Challenge
- JOKER - Automatic Wordplay Analysis
- LifeCLEF - Multimedia Retrieval in Nature
- LongEval - Longitudinal Evaluation of Model Performance

Campagnes d'évaluation

Autre

- NTCIR (Japon et Asie)
- FIRE (Inde)
- MediaEval (Multimedia / Europe)

Organisation

Objectifs en RI

Recherche d'Information

- Indexer et interroger une collection de documents
  $\rightarrow$ Développer un moteur de recherche
- Évaluer un moteur de recherche
- Comprendre les avancées récentes
  $\rightarrow$ deep learning

Organisation du cours (à titre indicatif)

- RI 1 (6 mars) – Indexation
- RI 2 (20 mars) – Modèle / Evaluation
- RI 3 (27 mars) – Page Rank / Learning to Rank
- RI 4 (3 avril) – Intervention Qwant
- RI 4 (17 avril) – Deep Learning pour la RI

Outils : Pyterrier, ElasticSearch

Evaluation

- Contrôle continu 50%
  - En RI : mini-projet (ré-implémentation de papier)
  - En TAL : (vu avec Nicolas Thome)
- Examen terminal 50%

La base

Notions de base

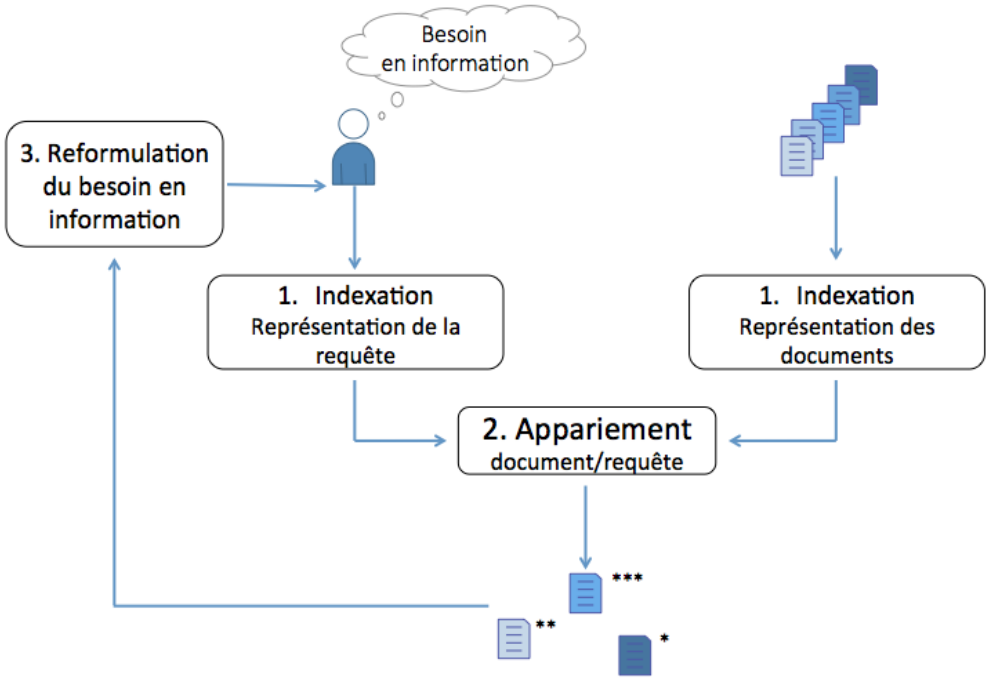
Document Texte, résumé, passage de texte, texte + structure (ex. balises HTML : titres, paragraphes, ...)...

Corpus Ensemble de documents textuels (statique ou dynamique), éventuellement liens entre documents.

Question Expression en texte "libre" formulée par l'utilisateur (ex."textmining", "je voudrais trouver des documents qui parlent de ...", paragraphes entiers, questions ambiguës "apple", "java", "jaguar"...)

- questions *navigationnelles* qui consistent à atteindre une page web particulière, connue par l'utilisateur (ex. "Sorbonne université").
- questions *transactionnelles* qui souhaitent réaliser des transactions ou bénéficier de services en ligne (ex. "Avion pas cher Toulouse-Paris").
- questions *informationnelles* qui ont pour objectif de chercher de l'information en rapport à un sujet donné, sans aucun a priori sur la source d'information (ex. "gaz à effet de serre").

Schéma général



- Processus en U avec 3 étapes :

  Indexation permet d'extraire le contenu d'un document dans un index

  Appariement permet de mettre en relation la collection de documents, indexée au préalable, avec la question, également pré-traitée, afin d'identifier les documents pertinents.

  Reformulation du besoin en information permet de redéfinir le besoin de l'utilisateur au fur et à mesure de la session de recherche.

Pertinence système vs. Pertinence utilisateur

- La pertinence est issue de la mise en correspondance de trois éléments :
  - La question
  - Le document
  - Le besoin utilisateur
    dépend de la tâche, du contexte, du temps, de la fraîcheur, ...
- **La pertinence système** : mesure algorithmique basée sur le calcul de l'adéquation entre la représentation de la question et celle de la collection de documents.
- **La pertinence utilisateur :** pertinence subjective que l'utilisateur aurait donné à chacun des documents.

Le compromis éternel de la RI

Définition

Rappel = % de documents pertinents *renvoyés* parmi tous ceux qui sont pertinents

Précision = % de documents pertinents *renvoyés* parmi ceux renvoyés

En appelant R l'ensemble des documents renvoyés, P les documents pertinents, on a

$$\textbf{Rappel} = \frac{|R \cap P|}{|P|} \text{ et } \textbf{Précision} = \frac{|R \cap P|}{|R|}$$

Le compromis éternel de la RI

Définition

Rappel = % de documents pertinents *renvoyés* parmi tous ceux qui sont pertinents

Précision = % de documents pertinents *renvoyés* parmi ceux renvoyés

En appelant R l'ensemble des documents renvoyés, P les documents pertinents, on a

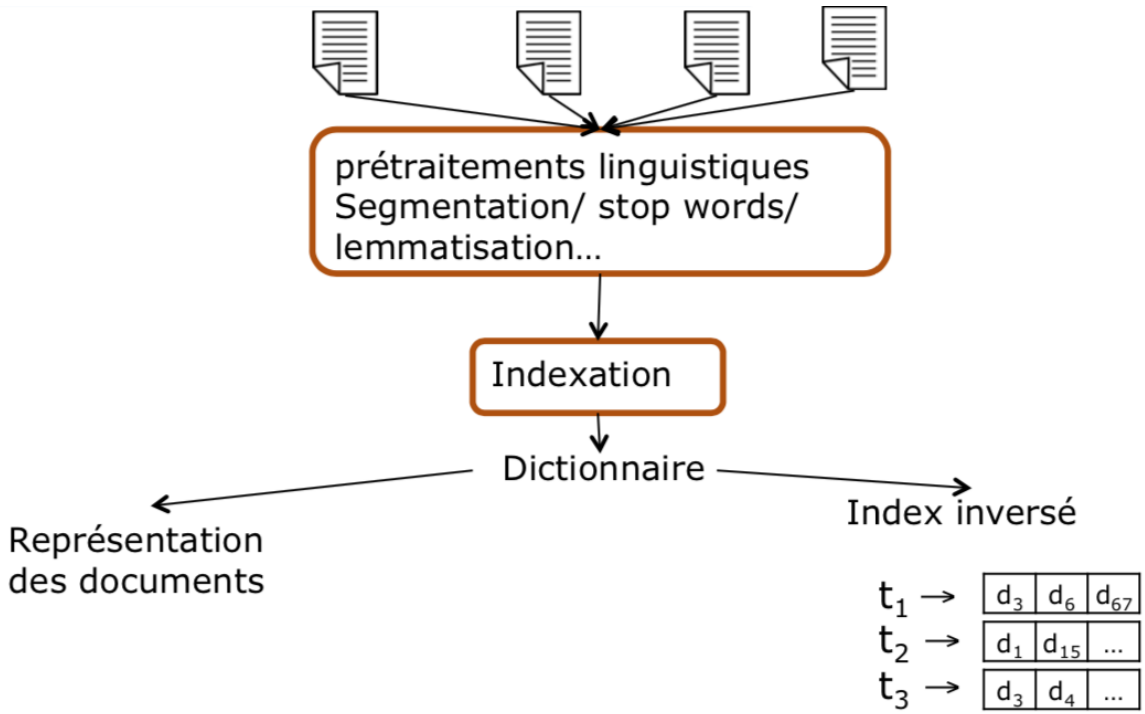
$$\textbf{Rappel} = \frac{|R \cap P|}{|P|} \text{ et } \textbf{Précision} = \frac{|R \cap P|}{|R|}$$

Rappel vs Précision

Si on veut augmenter le rappel, on risque de diminuer la précision ;
Si on veut augmenter la précision, on risque de diminuer le rappel ;

Pré-traitements

Chaîne d'indexation



- Indexation peut être manuelle, automatique, semi-automatique
- Elle peut aussi reposer sur un langage libre (issu du texte) ou contrôlé (lexiques, ressources sémantiques, ...)
- L'objectif est d'identifier la distribution des termes pour représenter les documents

Lois de distribution des termes pour les collections de documents

Loi de Zipf

- Stipule que la fréquence d'occurrence d'un mot est inversement proportionnelle à celle de son rang
- Principe du "moindre effort" : plus facile pour un auteur de répéter des mots que d'en chercher des nouveaux (Quelques mots communs représentent la plus grande partie des textes (stopwords))
- Le 1er mot est environ 2 fois plus fréquent que le 2nd qui est 2 fois plus fréquent que le 3e etc...
- Brown Corpus ($>$ 1 M mots)

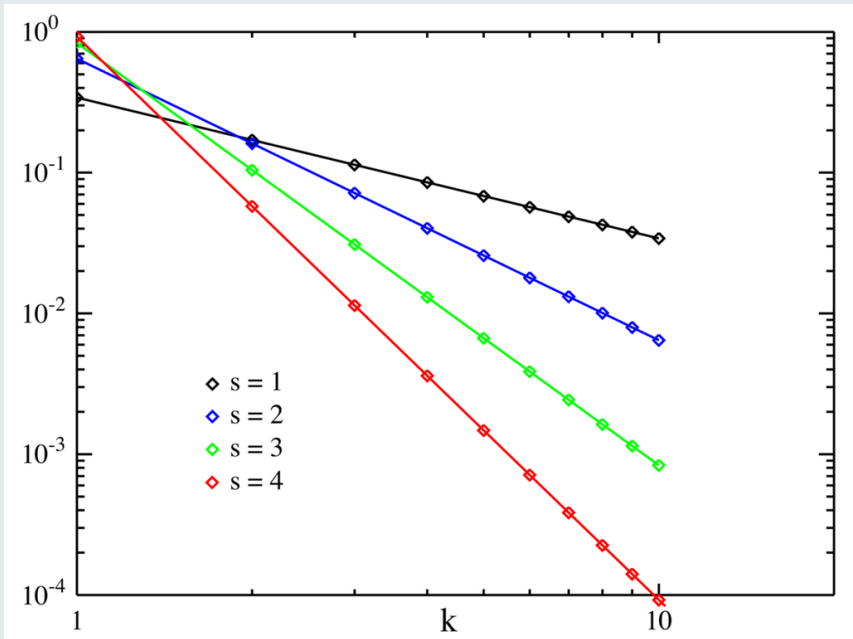
Mot	Rang	Occurrence	Fréquence
the	1	69971	7%
of	2	36411	3.5%
and	3	28852	2.6%

Lois de distribution des termes pour les collections de documents

Loi de Zipf

$$f(r, s, N) = \frac{\frac{1}{r^s}}{\sum_{n=1}^{N} \frac{1}{N}} \propto \frac{1}{r^s}$$

où r : rang, N : taille du corpus, s : paramètre du corpus



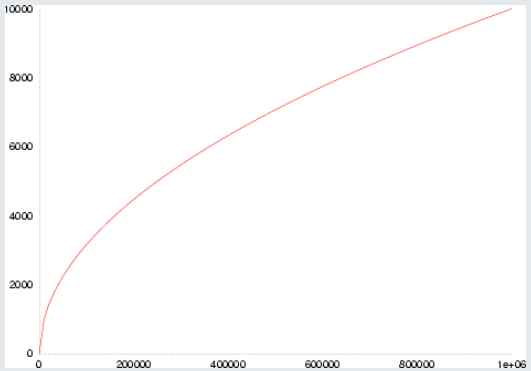
Analyse log fréquence vs. log rang. k = s et N = 10 (source Wikipedia)

Lois de distribution des termes pour les collections de documents

Loi de Heaps

Lien entre le *nombre de mots distincts* et le *nombre de mots*

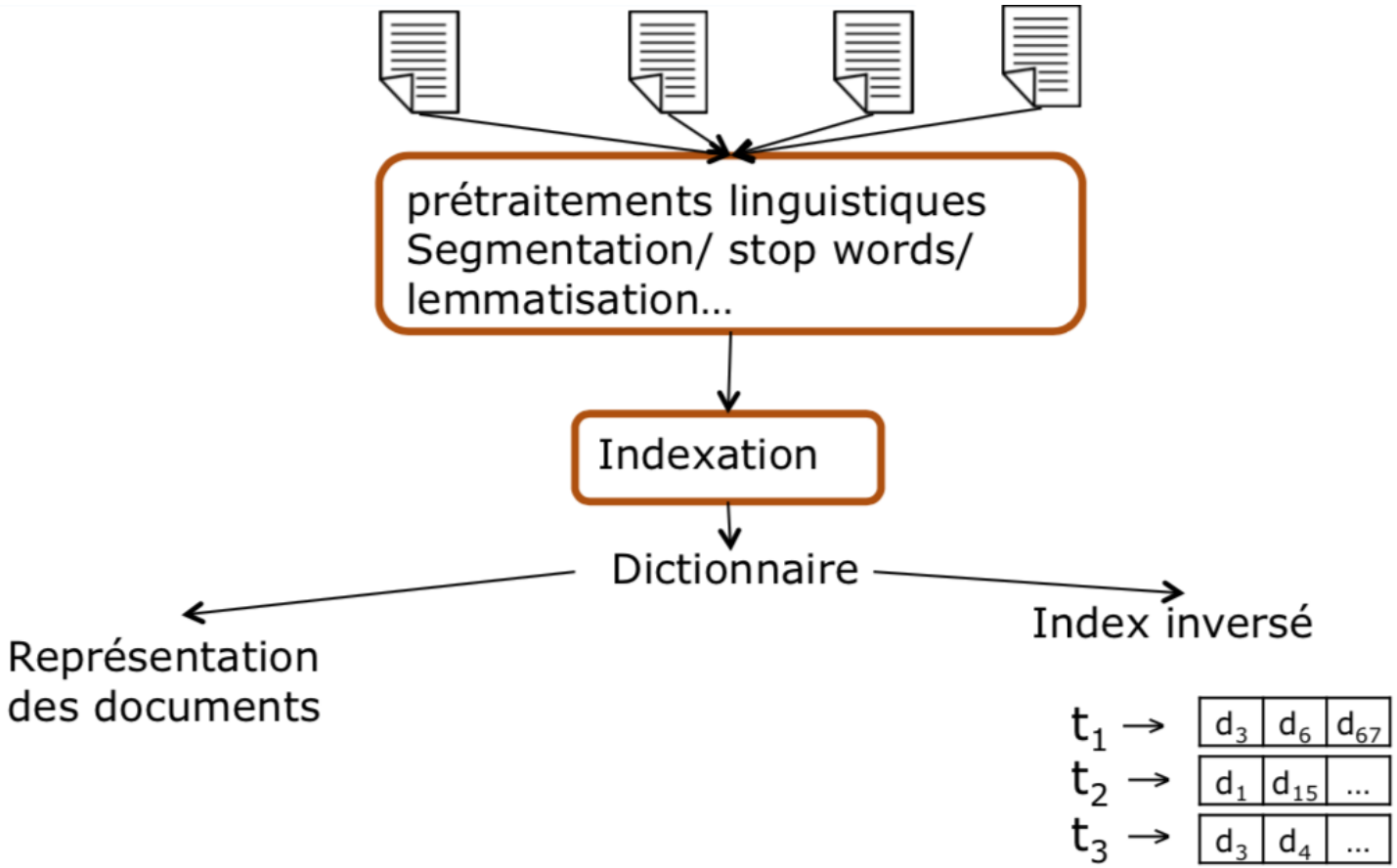
$$V = Kn^{\beta}$$



V : taille du vocabulaire, N : taille du texte, K et β : paramètres dépendant du texte (en Anglais : K entre 10 et 100 – β entre 0.4 et 0.6)

- L'index n'a pas borne supérieure (noms propres, erreurs de typos, etc.)
- Les nouveaux mots apparaissent moins fréquemment quand le vocabulaire croît (croissance sous-linéaire)

Chaîne d'indexation



Deux types d'index

- Index : représentation directe des documents
- Index inversé : représentation avec les termes pour point d'entrée.

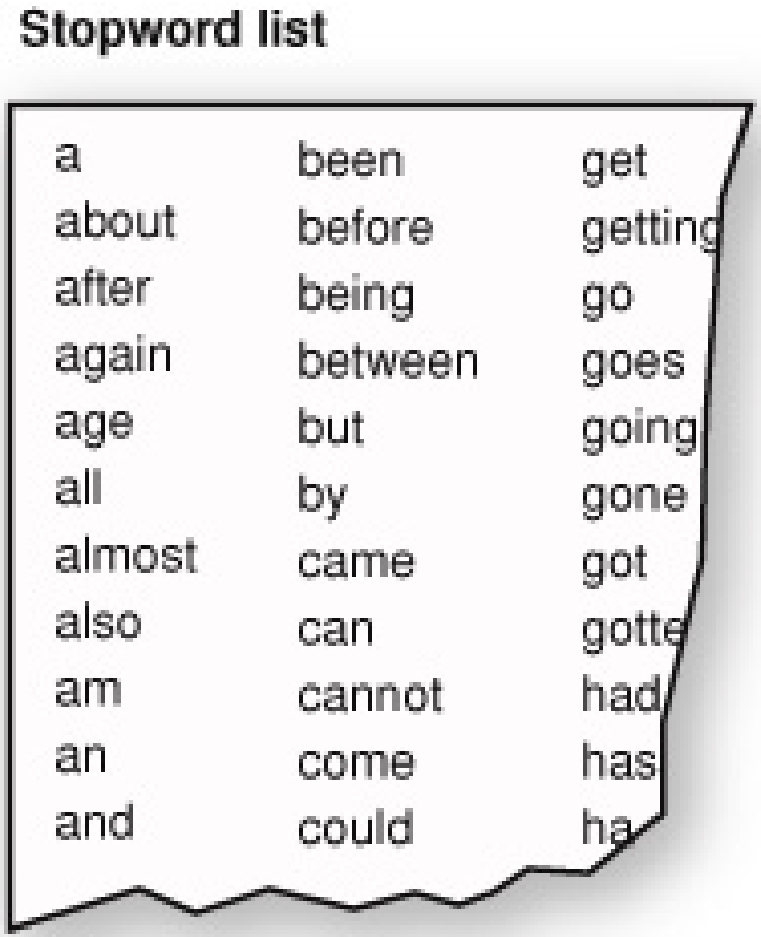
Prétraitement et représentation des textes - segmentation

- Identification des segments : mots simples (unigrammes), mots composés (bi-grammes), segments de 3 mots (tri-grammes), ..., N-grammes
- Analyse lexicale (segmentation – tokenisation)
- Conversion du texte en un ensemble de termes
  - Unité lexicale ou radical
  - Espaces, chiffres, ponctuations, etc
  - Dépend de la spécificité des langues traitées
    * ex. langues asiatiques (pas de signe de séparation) vs indo-européennes
    * Même pour les langues d'une même famille, nombreuses spécificités
      ex. aujourd'hui constitue un seul mot :
      Arbeiterunfallversicherungsgesetz (33 lettres) = Loi sur l'assurance des accidents du travail
  - Logiciel « TreeTagger » pour les langues indo-européennes

Prétraitement et représentation des textes - stopwords

- Quelles unités conserver pour l'indexation ? stopword/anti-dictionnaires
  - Les mots les plus fréquents de la langue "stop words" n'apportent pas d'information utile ex. prépositions, pronoms, mots « athématiques »,.. (peut représenter jusqu'à 30 ou 50% d'un texte)
  - Ces "stop words" peuvent être dépendants d'un domaine ou pas L'ensemble des mots éliminés est conservé dans un anti-dictionnaire (ex. 500 mots).
  - Les mots les plus fréquents ou les plus rares dans un corpus (frequency cut-off)
  - Les connaissances sémantiques permettent également d'éliminer des mots
  - Techniques de sélection de caractéristiques

Prétraitement et représentation des textes - stopwords



Prétraitement et représentation des textes - Normalisation

- Normalisation textuelle : transformations superficielles pour obtenir la forme canonique (ponctuations, casse, symboles spéciaux, accents, dates et valeurs monétaires)
- Normalisation linguistique :

  Racinisation regrouper les différentes variantes morphologiques d'un mot (cheval, chevalier, chevaux $\rightarrow$ cheva ; amusing, amusement, and amused $\rightarrow$ amus)

  Lemmatisation analyse linguistique ex. infinitif pour les verbes, singulier pour les noms (amusement, amusing, and amused $\rightarrow$ amuse)
- Regroupement de mots similaires au sens d'un critère numérique

Prétraitement et représentation des textes - Porter Stemmer

- Largement utilisé en anglais
- 5 phases de réduction des mots appliquées séquentiellement
- Règles de réécriture avec priorité d'application
- Exemple (Manning et al. 2008)
  - sses $\rightarrow$ ss : caresses $\rightarrow$ caress
  - ies $\rightarrow$ i : ponies $\rightarrow$ poni
  - ss $\rightarrow$ ss : caress $\rightarrow$ caress
  - s $\rightarrow$: cats $\rightarrow$ cat

Prétraitement et représentation des textes

- Représentations d'un document
  - Booléenne : présence/absence
  - Réelle : un indicateur numérique qui pondère le terme
  - Sélection de caractéristiques
  - Projections : réduction supplémentaire (SVD, ACP, NMF, Word2Vec, ...)
- Pondération des termes
  - Mesure l'importance d'un terme dans un document : Comment représenter au mieux le contenu d'un document ?
  - Considérations statistiques, parfois linguistiques
  - Loi de Zipf : élimination des termes trop fréquents ou trop rares
  - Facteurs de pondération
    * ex.tf (pondération locale), idf (pondération globale)
    * Normalisation :prise en compte de la longueur des documents, etc

Pré-traitement et représentation - Une méthode de référence : TF-IDF

- Term Frequency $tf(t_i, d)$: nombre occurrences de t_i dans le document d. Remarque : varie en fonction de la taille des documents. Si on double la taille des documents, tf double. Le document sera considéré comme plus pertinent.
- Inverse Document frequency *idf*

$$idf(t_i) = log\left(\frac{1+N}{1+df(t_i)}\right) \tag{1}$$

  - $df(t_i)$: nombre de documents contenant t_i
  - $idf(t_i)$: fréquence inverse, décroît vers 0 si t_i apparaît dans tous les documents
  - N : nombre de documents
- TF-IDF

$$x_i = tf(t_i, d) \times idf(t_i) \tag{2}$$

  - Il existe plusieurs variantes de ces poids (lissage, logarithme, ...)

Indexation et recherche

Modèles d'indexation - index

Index : représentation simple des documents

d1	(t1, n11) ; (t2,n12) ; ....
d2	(t1, n21) ; (t2,n22) ; ....
...	
dk	(t1, nk1) ; (t2,nk2) ; ....

Index inversé : point d'entrée par les mots

t1	(d1, n11) ; (d3,n13) ; ....
t2	(d4, n24) ; (d5,n25) ; ....
...	
tj	(d1, dj1) ; (d7 ; d72) ; ....

Modèles d'indexation - index inversé

- Index inversé : point d'entrée par les mots

Doc 1
I did enact Julius Caesar: I was killed i' the Capitol; Brutus killed me.

Doc 2
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:

term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

⟹

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

⟹

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
I	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Figure 1 – source : Manning et al. 2008

Modèles d'indexation - index inversé

- Index inversé : point d'entrée par les mots

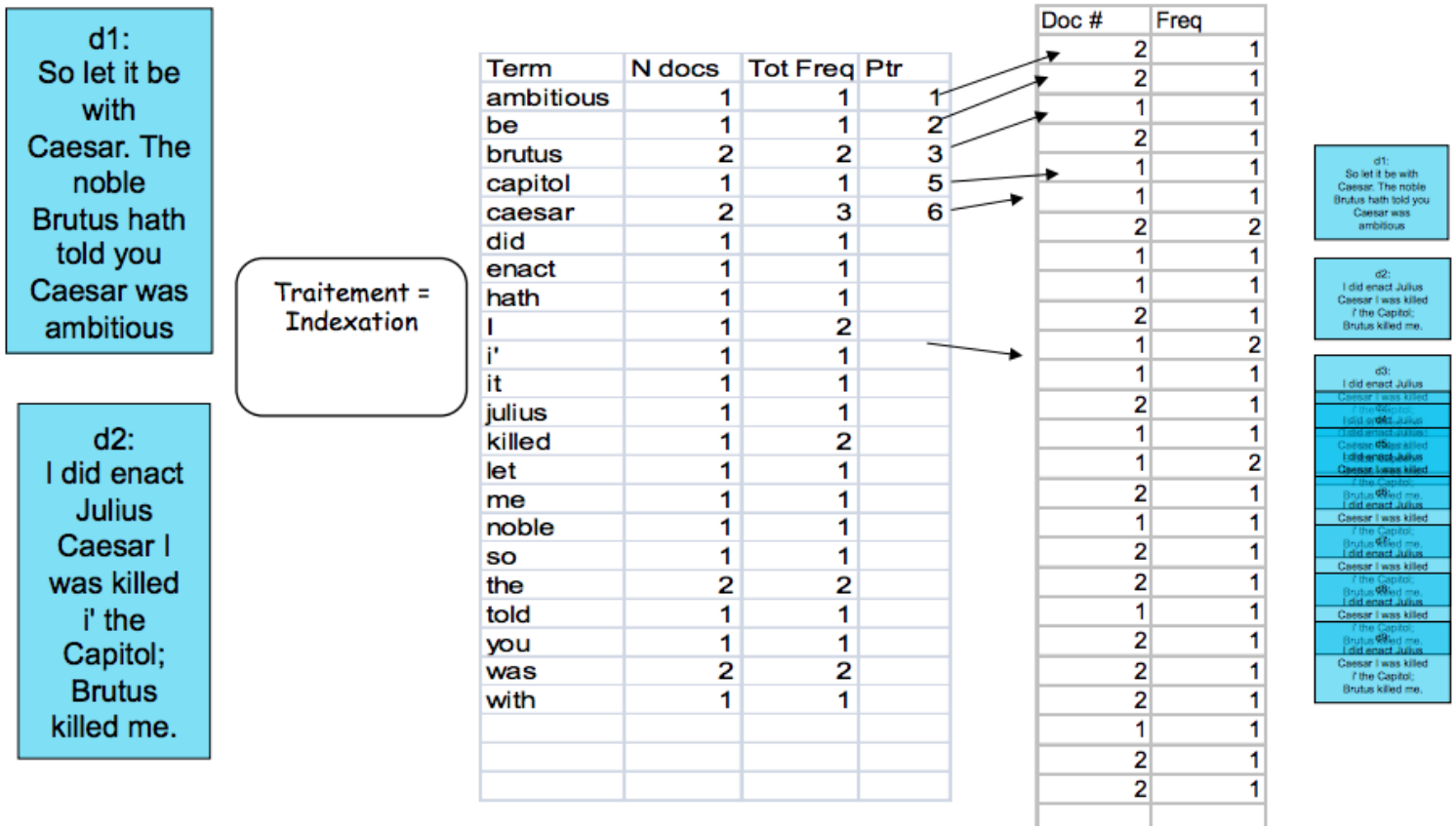


Figure 2 – source : Mohand Boughanem

Comment chercher rapidement ?

Problématique

- On a beaucoup de documents
- On cherche les K premiers

Comment chercher rapidement ?

Problématique

- On a beaucoup de documents
- On cherche les K premiers

DAAT vs TAAT

Deux principales stratégies

TAAT Term At A Time

DAAT Document At A Time

Comment chercher rapidement ?

Problématique

- On a beaucoup de documents
- On cherche les K premiers

DAAT vs TAAT

Deux principales stratégies

TAAT Term At A Time

DAAT Document At A Time

Notations

- $d_{t,i}$ et $w_{t,i}$ le $k$ème document (numéro) du $t$ème terme et son importance
- $w_{t,d}$ l'importance du terme t pour le document d
- c_t l'index courant pour le terme t

On suppose que

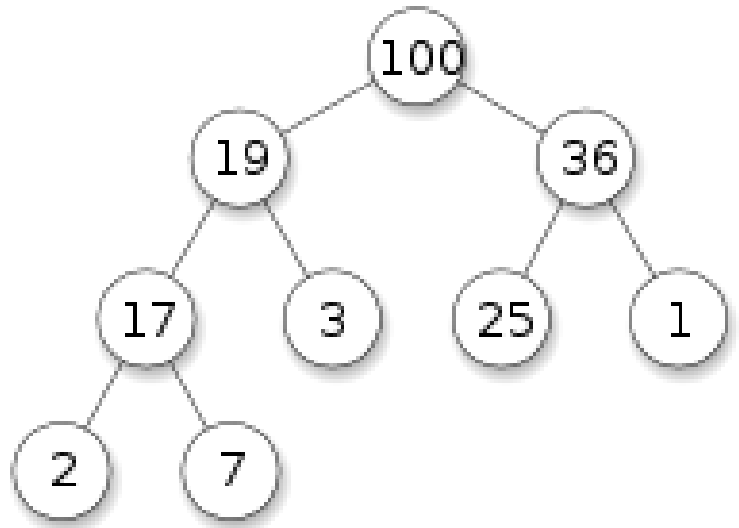
$$s(q,d) = \sum_{m \in q} w_{md}$$

Une structure essentielle : la structure Heap (tas)

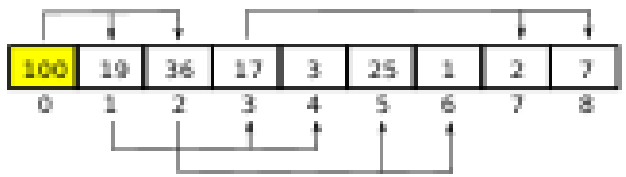
- Permet de trouver le maximum en $O(1)$
- Insertion/Remplacement en $\log(n)$

Cette structure permet de tenir à jour de manière efficace les K documents avec le plus grand score

Tree representation



Array representation



TAAT

Principe

- Pour chaque terme, les listes sont triées par importance : $w_{t,i} \geq w_{t,i+1}$
- Les termes de la questions sont triés par importance $q_1, \ldots, q_N$
- On parcourt l'index du terme q_n

À chaque étape n, on sait que le score d'un document est borné

$$s_n(q,d) = \sum_{j=1}^{n} w_{q_j,d} \leq s(q,d) \leq s_n(q,d) + \sum_{j=n+1}^{N} w_{q_j,1} = S_n(q,d)$$

- Le Kème document d_n^K (trié par s_n)

$=$ borne inférieure du score pour être dans le top K

- On peut "éliminer" tous les documents d tels que $s_n(q,d) < s_n(q,d_n^K)$

⊕ Très utile quand les termes ont de grandes différences d'importance

⊖ Peu performant quand le nombre de documents est très grand

DAAT

Principe

- Pour chaque terme, les listes sont triées par document : $d_{t,k} < d_{t,k+1}$
- On avance chaque curseur c_t de façon à parcourir l'ensemble des documents
- On ajoute un document au top-K que s'il est au-dessus d'une certaine valeur

- ⛔ Plus difficile de filtrer les "mauvais" candidats
- ➕ Plus efficace pour les grandes collections

DAAT : WAND (Weighted-AND)

SCIENCES SORBONNE UNIVERSITÉ

☞ Idée = trouver l'ID minimum d'un candidat

ligne 3 On trie les entrées par ID de document croissant

ligne 6 On cherche la première entrée telle que le score maximum accumulé soit supérieur à θ

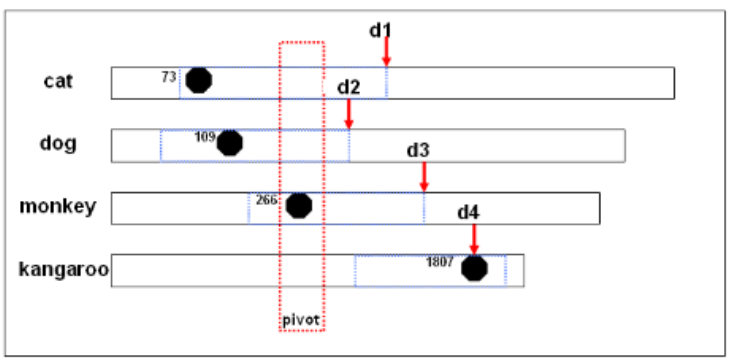


Figure 5: An example showing how *GetNewCandidate()* works. Assume 266 is the pivot and it fails to make it into the top results. In this case, we enable better skipping by choosing min$(d1, d2, d3, d4)$ as the next possible candidate, instead of $266 + 1$

From Ding et Suel (2011)

```
1. Function next(θ)
2.   repeat
3.     /* Sort the terms in non decreasing order of
        DID */
4.     sort(terms, posting)
5.     /* Find pivot term - the first one with accumulated
        UB ≥ θ */
6.     pTerm ← findPivotTerm(terms, θ)
7.     if (pTerm = null) return (NoMoreDocs)
8.     pivot ← posting[pTerm].DID
9.     if (pivot = lastID) return (NoMoreDocs)
10.    if (pivot ≤ curDoc)
11.      /* pivot has already been considered, advance
          one of the preceding terms */
12.      aterm ← pickTerm(terms[0..pTerm])
13.      posting[aterm] ← aterm.iterator.next(curDoc+1)
14.    else /* pivot > curDoc */
15.      if (posting[0].DID = pivot)
16.        /* Success, all terms preceding pTerm belong
            to the pivot */
17.        curDoc ← pivot
18.        return (curDoc, posting)
19.      else
20.        /* not enough mass yet on pivot, advance
            one of the preceding terms */
21.        aterm ← pickTerm(terms[0..pTerm])
22.        posting[aterm] ← aterm.iterator.next(pivot)
23.    end repeat
```

Broder et al. (2003)

Modèles neuronaux

Modèles neuronaux

Deux grandes familles de modèles “first-stage” neuronaux :

Dense Un document est un vecteur parcimonieux de $\mathbb{R}^d$

Sparse Un document est un vecteur parcimonieux de $\mathbb{R}^n$

- Modèles denses : Approches par clustering (ex. FAISS de Johnson, Douze et Jégou (2019)) – le but est de trouver $i = \text{argmin}_i ||x - x_i||$
- Modèles parcimonieux : les algorithmes de type DAAT sont les plus étudiés à l’heure actuelle Mackenzie, Mallia et Moffat 2022 – et de “vieux” algorithmes comme MaxScore Turtle et Flood 1995

References

Ressources

- Ressources pédagogiques
  - Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008.
  - Massih-Reza Amini, Eric Gaussier, *Recherche d'information, Applications, modèles et algorithmes*, Eyrolles 2013-2018
  - W. Bruce Croft, Donald Metzler, Trevor Strohman, *Search Engines Information Retrieval in Practice*, Addison Wesley, 2009
- Ressources scientifiques
  - Conférences : SIGIR, CIKM, ECIR, ICTIR, WSDM, WWW, CORIA
  - Journaux : ACM-TOIS, JASIST, IP&M, JIR

Références I


Broder, Andrei Z. et al. (3 nov. 2003). « Efficient Query Evaluation Using a Two-Level Retrieval Process ». In : *Proceedings of the Twelfth International Conference on Information and Knowledge Management*. CIKM '03. New York, NY, USA : Association for Computing Machinery, p. 426-434. ISBN : 978-1-58113-723-1. DOI : 10.1145/956863.956944. URL : http://doi.org/10.1145/956863.956944 (visité le 03/07/2022).


Ding, Shuai et Torsten Suel (juill. 2011). « Faster top-k document retrieval using block-max indexes ». In : *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. SIGIR '11. New York, NY, USA : Association for Computing Machinery, p. 993-1002. ISBN : 978-1-4503-0757-4. DOI : 10.1145/2009916.2010048. URL : http://doi.org/10.1145/2009916.2010048 (visité le 03/07/2022).


Johnson, Jeff, Matthijs Douze et Hervé Jégou (2019). « Billion-scale similarity search with GPUs ». In : *IEEE Transactions on Big Data* 7.3, p. 535-547.


Mackenzie, Joel, Antonio Mallia et Alistair Moffat (2022). « Accelerating Learned Sparse Indexes Via Term Impact Decomposition ». en. In : *Findings of the Association for Computational Linguistics : EMNLP 2022*.


Turtle, Howard et James Flood (nov. 1995). « Query evaluation : Strategies and optimizations ». en. In : *Information Processing & Management* 31.6, p. 831-850. ISSN : 0306-4573. DOI : 10.1016/0306-4573(95)00020-H. URL : https://www.sciencedirect.com/science/article/pii/030645739500020H (visité le 22/12/2022).