Techniques d'indexation et recherche multimédia

Chap2- Indexation des documents texte

Amira BELHEDI amira.belhedi@istic.ucar.tn
Fadoua MHIRI fadoua.mhiri@istic.ucar.tn
Wafa ABID wafa.abid@istic.ucar.tn

ISTIC 2021-2022

Plan

- Modes d'indexation
  - manuelle
  - Automatique
  - semi-automatique
- Etapes de l'indexation textuelle
  - Extraction
  - Etalage
  - Normalisation
  - Pondération
- Construction du fichier inversé
- Indexation textuelle sémantique

Indexation

- Indexation: comprend deux parties
  1. Extraction des descripteurs à partir des documents
     - descripteur représente le contenu d'un document ou d'une requête
     - doit refléter au mieux le contenu
  2. Représentation de cette information.

Indexation textuelle

- Utiliser des mots-clés (texte)
  - termes significatifs se trouvant dans le document / la requête

Modes d'indexation

- Trois modes avec lesquels l'indexation peut être réalisée
  - Manuelle: analyse du document par un spécialiste du domaine ou par un documentaliste
  - Automatique: processus entièrement automatisé
  - Semi-automatique: combinaison des deux modes
- Basée sur
  - Un langage contrôlé (lexique/thesaurus/ontologie/réseau sémantique)
  - Un langage libre (éléments pris directement des documents)

Pourquoi l'indexation?

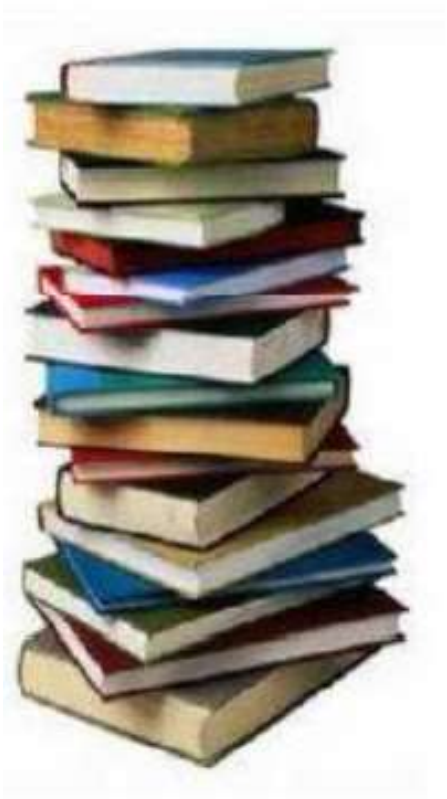
- L'objectif d'un SRI est de retrouver les documents qui « parlent de » la requête
- Approche naïve: parcours séquentiel de la requête et du document
  - Considère le document comme une liste de mots
  - Considère la requête comme une liste de mots
  - Sélectionne les documents qui contiennent les mêmes mots que la requête (par comparaison des deux listes)

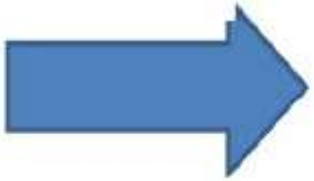
➔ Approche couteuse + pouvoir d'expression des besoins limité

Indexation textuelle automatique

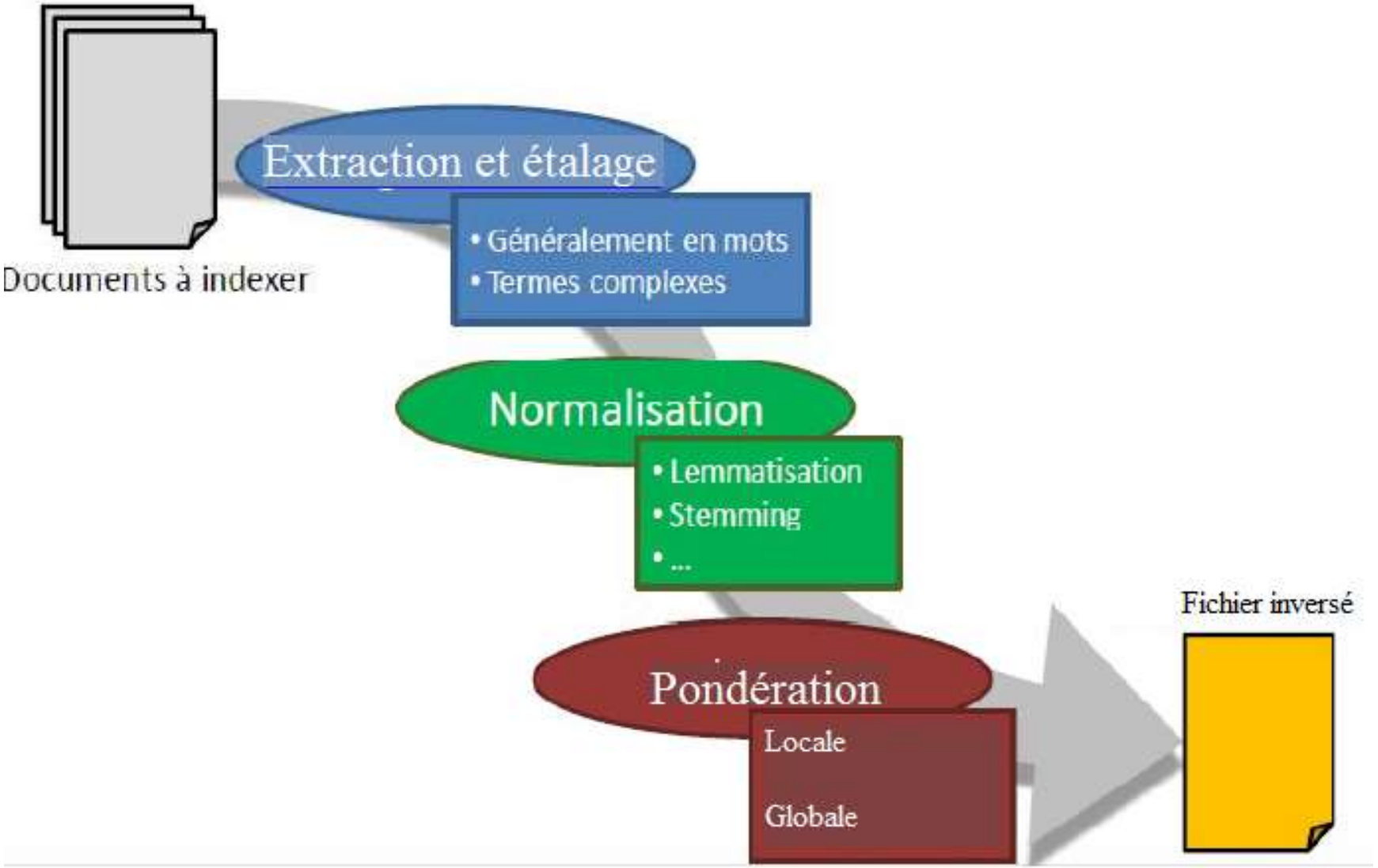
- Le but de l'indexation textuelle automatique: « *transformer des documents en substituts capables de représenter le contenu de ces documents* » [Salton et MCGill, 1983]
- L'indexation est un processus qui permet de
  - Trouver les concepts importants dans un document
  - Ensuite, de créer une représentation interne du document (ou de la requête) à partir des concepts trouvés
    - Dj ➔ {Ci tel que Ci représente Dj}
  - **Exemples:**
    - D1 ➔ {C, Java}
    - D2 ➔ {UML, réseau}

Indexation textuelle: fichier inversé





Chaine de l'indexation textuelle



Etapes de l'indexation textuelle

- Extraction
- Etalage
- Normalisation des termes
- Pondération

Extraction

- Comment reconnaitre les mots dans un texte
- exemple de texte

il	fait	beau	le	lundi

- Définir les termes suivants:
  - texte / mot/ séparateur /terme

Extraction

- Extraction des mots (segmentation ou tokenisation): séparer le texte en mot
  - Texte: succession de symboles ou caractères: lettre, ponctuation, opérateur mathématique
  - Séparateur: exemples: espace, caractère de ponctuation, etc.
  - Mot: suite de caractères comprise entre deux séparateurs
  - Terme: mot ou groupe de mots

Extraction

- Unité (entité) d'indexation:
  - terme jugé valide pour représenter le contenu du document➔ candidat pouvant être utilisé pour l'indexation
- Descripteur:
  - terme utilisé effectivement pour l'indexation ➔ unité n'est pas forcement utilisée pour l'indexation

Extraction

- Exemple d'unité d'indexation
  - C++ et CPP
  - Voiture et automobile
- Descripteur extrait
  - CPP
  - voiture
- Un descripteur est choisi comme représentant d'un ensemble d'unités

Etalage

- Deux solutions sont possibles suite à l'extraction d'un ensemble de termes
  - Solution1: considérer tous les descripteurs extraits
    - Nombre élevé + efficacité réduite
  - Solution2: considérer un sous ensemble des descripteurs en fonction des critères suivants:
    - Informativité: ne retenir que les termes qui reflètent le contenu
    - Réduction de l'espace de stockage: réduction de la taille d'index
    - Augmentation de la performance: éviter le temps de calcul énorme induit par un nombre élevé d'index

Exercice 1

- soit les trois documents et leurs contenus respectifs:
  - D1: le petit chat est mort de faim et de froid
  - D2: le gros chat est rentré à la maison
  - D3: la maison résiste au froid
- et la requête utilisateur suivante:
  - Q: le froid est dangereux
- Considérant la mesure de similarité S définie comme suit: si un terme de la requête est présent dans D ➔ le score est incrémenté de 1,
- Calculer le score de chaque document et ordonner les documents en se basant sur le score calculé

Corrigé de l'exercice1

- Calcul du score de chaque document
  - S(D1) =3,
  - S(D2)=2,
  - S(D3)=1

- D2 est mieux classé que D3?
- Est-ce D2 a plus de lien avec Q que D3?
- Est-ce que ce classement est erroné, si oui, donner la cause

Corrigé de l'exercice1

- D2 est mieux classé que D3, cependant, D2 n'a qu'un lien avec Q. ➔ classement erroné

- Ce classement erroné induit par la présence des mots vides
  - Donner la liste des mots vides ?

➔Solution: éliminer les mots vides

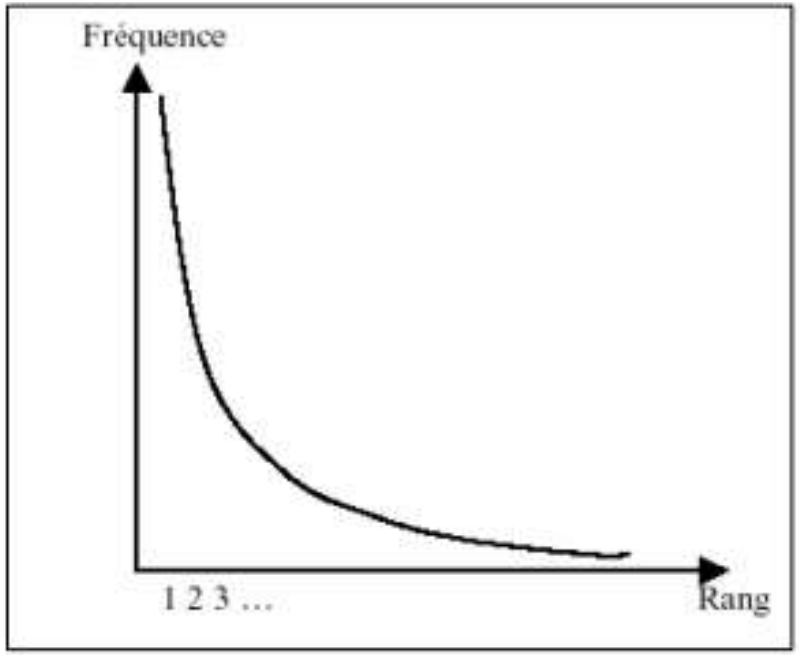
Etalage

- Quand on fait la statistique d'occurrence, on s'aperçoit que les mots les plus fréquents sont des mots fonctionnels (ou mots outils, mots vides).
  - En français, les mots "de", "un", "les", etc. sont les plus fréquents.
  - En anglais, ce sont "of", "the", etc.
- Ce phénomène n'est pas étrange si on connaît la loi de Zipf.

Loi de Zipf

- Si on classe les mots dans l'ordre décroissant de leur fréquence, et on leur donne un numéro de rang (1, 2, …), alors: Rang * fréquence ≈ constante
- La distribution de mots suit la courbe suivante:

Rang	Mot	Fréquence	Rang* Fréquence
1	the	69 971	69 971
2	of	36 411	72 822
3	and	28 852	86 556
4	to	26 149	104 596
5	a	23 237	116 185
6	in	21 341	128 046
7	that	10 595	76 165



Loi de Zipf

- Une idée est de garder les termes "utiles" :
  - ni trop rares: prennent de la place en mémoire pour rien
  - ni trop présents (pas discriminants)…

➔choix difficile

Etalage

- Le but est de maintenir un minimum de termes tout en préservant le contenu

1. Les mots outils: les mots associés à la langue du document et ne possède pas de sens, tels que:
   - En Français:
     - Les propositions: à, de, dans, etc.
     - Les articles: le, la, les, des, etc.
     - Les auxiliaires: être et avoir
     - Les pronoms: lui, mien, je, tu etc.
   - Anglais : the, or, a, you, I, us, …

- Attention à :
  - US : « USA » ➔ A ici est significative
  - a de (vitamine a)

Etalage

2. Les mots fonctionnels: spécifiques au domaine du corpus, exemple:
   - Le terme Ordinateur dans le corpus d'information
   - Le terme Loi dans le corpus de droit

- Pour ces deux cas, on utilise une liste appelée stoplits ou anti-dictionnaire, cette liste contient les termes qui ne sont pas candidats pour indexer les documents,

Normalisation

- Variation morphosyntaxique d'un mot
  - Flexion
    - Verbale: montrer, montreras
    - Nominale: cheval, chevaux
  - Dérivation
    - Penser + able = pensable
  - Composition
    - Pomme + de + terre
- Objectifs de la normalisation
  - Grouper les termes «similaires »: rassembler les différentes variantes morphosyntaxiques d'un mot autour d'un mot commun
  - Diminuer la taille des index

Normalisation

- Prétraitement du texte (analyse lexicale): le texte est découpé en lexèmes
  - Analyse morphosyntaxique: attribuer une catégorie grammaticale à chaque lexème identifié
  - Désambiguïsation: levée l'ambiguïté des lexèmes qui ont plus d'une catégorie grammaticale
  - Lemmatisation
  - Racinisation ou troncature (stemming)

Lemmatisation

- «Lemmatisation» : enlever les variations flexionnelles des mots afin de les ramener sous leur forme lemmatisée ou encyclopédique

  ➔Correspond à un mot réel de la langue

- Exemples:
  - verbe ➔ verbe à l'infinitif
  - mot ➔ mot au masculin singulier

Troncature

- Tronquer les mots à X caractères
  - Tronquer plutôt les suffixes
  - Exemple troncature à 7 caractères
    - économiquement ➔ économi

• Quelle est la valeur optimale de X ? : 7 caractères pour le Français ?

Racinisation

- Forme morphologique d'un mot

- La racine d'un mot correspond à la partie du mot restante une fois que l'on a supprimé son préfixe et son suffixe, à savoir son radical.
- Connue sous le nom de stemme d'un mot
- La racine ne correspond généralement pas à un mot réel de la langue.
- Exemples:
  - économie, économiquement, économiste ⇨ économ
  - retrieve, retrieving, retrieval, retrieved, retrieves ⇨ retriev

Racinisation

- Un des algorithmes de racinisation les plus connus: Algorithme de Porter:
  - Pour les mots en anglais uniquement
  - Comprend 50 règles qui sont appliquées en 5 étapes

- Les règles sont exprimées sous la forme:

  (condition) S1 → S2

  - c.à.d. si un mot se termine par S1 + son préfixe satisfait la condition alors le suffixe S1 est remplacé par S2

Règles de l'algorithme de Porter

- Exemple1:

(m>0) EED → EE

- m pour un «stem» est le nombre de séquences de VC (voyelle suivie d'une consonne)

- Exemples
  - (tree, by) → m=0
  - (trouble,oats, trees, ivy) → m=1
  - (troubles, private) → m=2

Règles de l'algorithme de Porter

- Exemple1:

(m>0) EED → EE

  - feed → feed
  - agreed→ agree

- Exemple2:

(*v*) ED →

(*v*) c.à.d. le préfixe contient une voyelle

  - plastered → plaster
  - bled → bled

Règles de l'algorithme de Porter

- Exemple3

  s ⟶

  (*v*) ed ⟶

  (*v*) ing ⟶

  (m>1) er ⟶

  (m>1) e ⟶

Mot initial	Mot tronqué
engineered	engin
engineer	engin
engineers	engin
informing	inform
computer	comput
computing	comput

Remarque:

- L'algorithme de Porter Comprend 50 règles qui sont appliquées en 5 étapes (il faut les appliquer étape par étape)

Exercice2: algorithme de Porter

- On considère le texte suivant

> marketing strategies carried out by U.S. companies for their agricultural chemicals, report predictions for market share of such chemicals, or report market statistics for agrochemicals, pesticide, herbicide, fungicide, insecticide, fertilizer, predicted sales, market share, stimulate demand, price cut, volume of sales

1. Donner la liste des mots vides (stopList ou anti-dictionnaire)
2. Récupérer le fichier « RèglesDePorter_Ex2_Chap2» qui contient les règles de l'algorithme de Porter
   a. Donner le résultat obtenu après l'application de l'algorithme de Porter et la suppression des mots vides ()
   b. Indiquer pour chaque terme le nombre de ces occurrences.

Corrigé de l'exercice 2

- StopList {out by for their such of or }
- Résultat obtenu:

Market 4
strateg 1
carr 1,
compan 1
US 1
agricultur 1
chemic 2
report 2
predict 2
share 1
statist 1
agrochem 1
pesticid 1
herbicid 1
fungicid 1
insecticid 1
fertil 1
sale 2
stimul 1
demand 1
price 1
cut 1
volum 1

Pondération

- Dans un document, les termes ne possèdent pas la même importance
  - Comment caractériser l'importance d'un terme dans un document ?
- Associer un poids à chaque terme reflétant son importance
  - Les termes importants doivent avoir un poids élevé?

Pondération

- L'importance d'un terme dans un document dépend de:
  - L'importance du terme dans le document (pondération locale)
  - L'importance du terme dans la collection (pondération globale)

- Remarque: La taille des documents est différente ➔ il faut normaliser par rapport à la taille du document

Pondération locale

- Cherche à mesurer la représentativité d'un terme au sein d'un document

- Plus un document contient d'occurrence d'un terme, plus il est important dans la description du document

➔ Mesure TF (Term Frequency)

Pondération locale

- Plusieurs formules de la mesure TF
- Exemples:
  - **TF mesure simple**: fréquence du terme Ti dans le document Dj

$$TFi,j = f(Ti, Dj)$$

  - **TF mesure normalisée**: normalisation de la fréquence simple par rapport à la taille du document

$$TFi,j = \frac{f(Ti,Dj)}{1{,}5x\frac{longueur\_doc\_Dj}{longueur\_moy\_doc} + f(Tj,Dj) + 0{,}5}$$

$longueur\_doc\_Dj$ = nombre de descripteurs du document Dj

Pondération globale

- Cherche à mesurer la représentativité d'un terme au sein de la collection

- Intuition 2: les termes très fréquents dans tous les documents ne sont pas si importants (ils sont moins discriminants)

➔ mesure IDF (Inverse Document Frequency)

Pondération globale

- Plusieurs formules de la mesure IDF
- Exemples:
  - **IDF Mesure simple**

$$IDFi = \frac{1}{Ni}$$

  - **IDF Mesure en log**

$$IDFi = \log_{10}\left(\frac{N}{Ni}\right)$$

    - Ni : nombre de documents ou le terme Ti apparait
    - N est le nombre total de documents dans le corpus

Pondération

- Pondération des termes est la combinaison de la pondération locale et globale

➔ Poids (weight) d'un terme Ti dans le document Dj

- Poids mesure simple

$$Wi,j = TFi,j \; x \; IDFi$$

Exercice 3

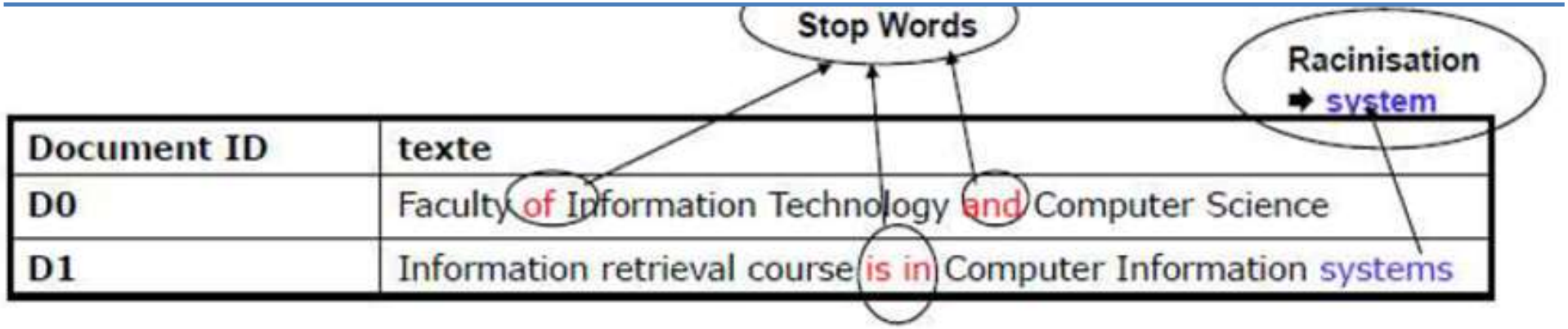
- Soit les deux documents D1 et D2 et leurs contenus respectifs:

Document ID	Texte
D1	Faculty of information technology and computer science
D2	Information retrievel course is in computer information systems

Donner le résultat à chaque étape de l'indexation

1. Extraction
2. Etalage (donner liste des mots vides
3. Normalisation (utiliser la lemmatisation lemmatisation)
4. Pondération:
   - Locale: pour chaque document, calculer TF (Term Frequency), utiliser la mesure simple
   - Globale: pour chaque terme calculer IDF (Inverse Document Frequency), utiliser la mesure en log
   - Donner le poids de chaque terme de la collection

Corrigé exercice 3



Document ID	texte
D0	Faculty of Information Technology and Computer Science
D1	Information retrieval course is in Computer Information systems

Document ID	TF
D_0	
Faculty	1
Information	1
Technology	1
Computer	1
Science	1
D_1	
Information	2
Retrieval	1
Course	1
Computer	1
Systems	1

Ni : le nombre de documents ou Ti apparait

Term ID	Term	Ni	IDF
t1	Faculty	1	$\log_{10} 2/1=0.301$
t2	Information	2	$\log_{10} 2/2=0$
t3	Technology	1	$\log_{10} 2/1=0.301$
t4	Computer	2	0
t5	Science	1	0.301
t6	Retrieval	1	0.301
t7	Course	1	0.301
t8	System	1	0.301

Corrigé exercice 3

- Calcul du poids des différents termes

$$Wi,j = TFi,j \; x \; IDFi$$

	Terme ID							
	T1	T2	T3	T4	T5	T6	T7	t8
D0	1*.301 = .301	1*0=0	1*.301 = .301	1*0=0	1*.301 = .301	0*.301 =0	0*.301 =0	0*.301 =0
D1	0*.301 =0	2*0=0	0*.301 =0	1*0=0	0*.301 =0	1*.301 = .301	1*.301 = .301	1*.301 = .301

Fichier inversé

- Une fois les documents indexés :
  - chaque document aura donc un descripteur (une liste de mots souvent simples) ➔Sac de mots (Bag of Words)
  - Ces termes sont ensuite stockés dans une structure appelée fichier inversé

Organisation du fichier inversé

Dictionnaire

Terme	Nb Doc	FrqTotal	Ptr
Ambitious	2	5	1
Brutus	2	4	3
Capitol	1	3	5
…..			

Posting simple

doc	Freq
doc1	3
doc2	2
doc1	1
doc3	3
doc2	1

- Liste triée par ordre alphabétique

Position du terme dans le document (important pour la recherche d'expressions)

Posting riche

doc	Freq	position	balise
doc1	3	1, 4, 3	1, 5
doc2	2	1	
doc3	2	3	

Balises (title, body, anchor, ..)

Structure du Fichier Inversé

d2:
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

d1:
I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Traitement = Indexation

Term	N docs	Tot Freq	Ptr
ambitious	1	1	1
be	1	1	2
brutus	2	2	3
capitol	1	1	5
caesar	2	3	6
did	1	1	
enact	1	1	
hath	1	1	
I	1	2	
i'	1	1	
it	1	1	
julius	1	1	
killed	1	2	
let	1	1	
me	1	1	
noble	1	1	
so	1	1	
the	2	2	
told	1	1	
you	1	1	
was	2	2	
with	1	1	

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

d1:
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

d2:
I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

d3:
I did enact Julius

d4:

d5:
I did enact Julius Caesar I waskilled

d6:
I did enact Julius Caesar I waskilled

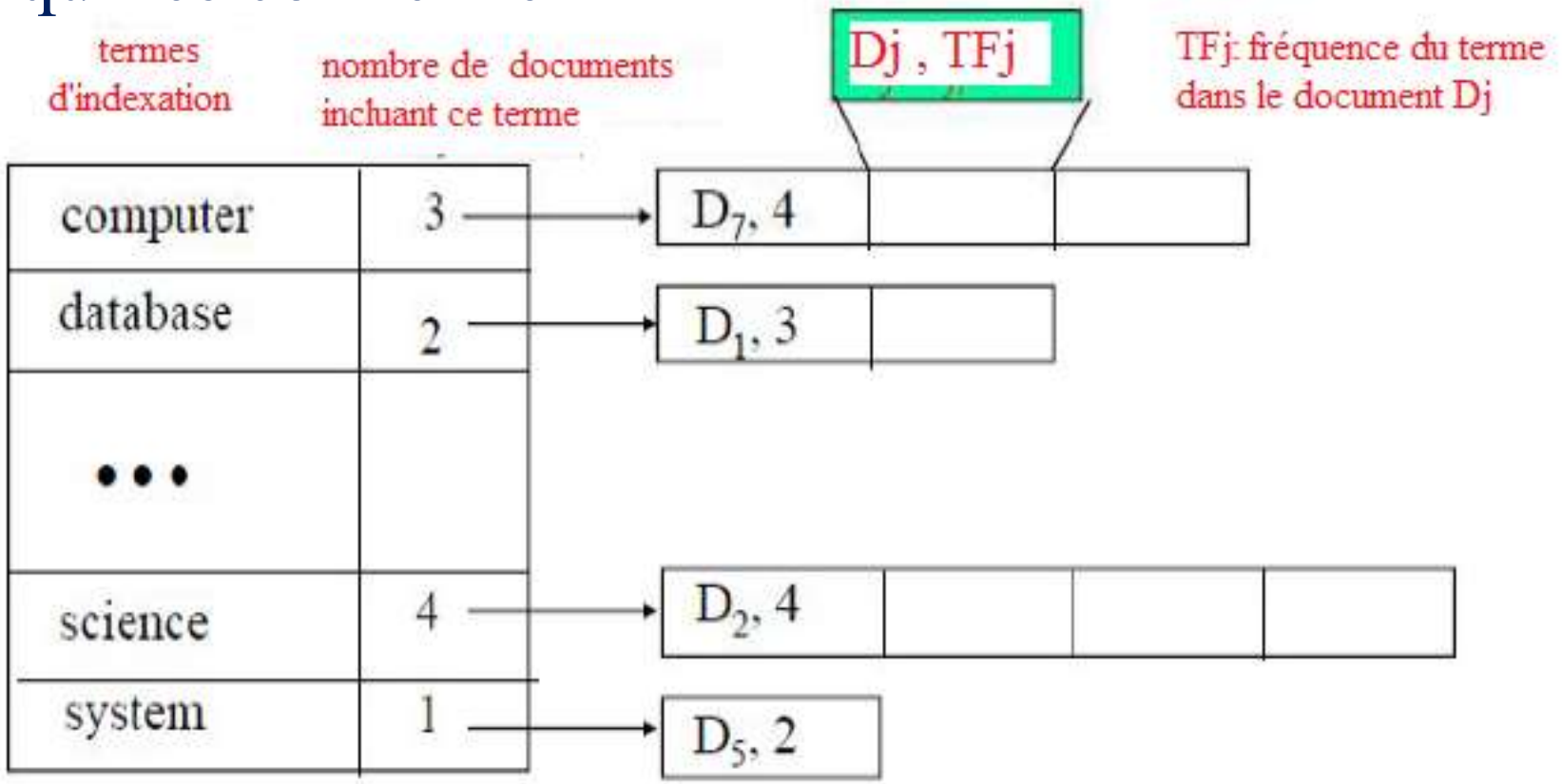
d7:
I did enact Julius Caesar I waskilled

d8:
I did enact Julius Caesar I waskilled

d9:
I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Structure du Fichier Inversé

- Un fichier inversé associe des index aux documents qui les contiennent



Démarche de construction d'un fichier inversé

- La construction d'un fichier inversé est une «étape importante

- Elle peut prendre énormément de temps

Extraction des mots de chaque document

- Extraire les termes de chaque document dans un fichier (1 fichier par document ou 1 fichier pour plusieurs documents)

Doc 1

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

Doc 2

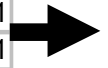
So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	Doc #
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

Tri du fichier termes-documents (1/2)

- Trier le fichier par ordre alphabétique des termes et par document

Term	Doc #
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	Doc #
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

Tri du fichier termes-documents (2/2)

- Pour chaque terme,
  - Calculer la fréquence d'apparition dans chaque document

- Pour chaque terme,

➔ On dispose de la liste de documents qui le contient

➔ Le nombre de documents comportant ce terme

Term	Doc #
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
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
i'	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1
	52/70	40

Construction du dictionnaire et du « posting »

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

Dictionnaire

Term	N docs	Tot Freq	Ptr
ambitious	1	1	1
be	1	1	2
brutus	2	2	3
capitol	1	1	5
caesar	2	3	6
did	1	1	
enact	1	1	
hath	1	1	
I	1	2	
i'	1	1	
it	1	1	
julius	1	1	
killed	1	2	
let	1	1	
me	1	1	
noble	1	1	
so	1	1	
the	2	2	
told	1	1	
you	1	1	
was	2	2	
with	1	1	

Posting

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

Fichier inversé

- Remarque:
  - Dans un fichier inversé on peut ajouter d'autres informations selon le modèle de recherche adopté
  - Exemple:
    - Pour un modèle pondéré (booléen ou vectoriel): ajout du poids Wi,j
    - Pour un modèle vectoriel: ajout de TF et IDF
- Plus de détails dans le chapitre 4

Exercice 4

- Construire le fichier inversé correspondant à cette collection de documents

Document ID	Texte
D1	Faculty of information technology and computer science
D2	Information retrievel course is in computer information

- Remarque:
  - Utiliser le résultat des étapes d'extraction, étalage et normalisation (slide 44)

Corrigé de l'exercice 4
Fichier inversé

Dictionnaire

Terme	Nb doc	Total Freq	Ptr
Computer	2	2	1
Course	1	1	3
Faculty	1	1	4
Information	2	3	5
Retrieval	1	1	7
Science	1	1	8
System	1	1	9
Technology	1	1	10

Posting simple

Doc #	Freq
1	1
2	1
2	1
1	1
1	1
2	2
2	1
1	1
2	1
1	1

Répondre à une demande

- Pour répondre à une requête, on doit définir le modèle de recherche adopté

- Dans cet exemple, on va utiliser le modèle booléen (Vrai ou Faux): le mot existe ou n'existe pas

➔ ce modèle sera détaillé dans le chapitre 4

Répondre à une demande
Exemple requête avec un seul mot

- Modèle de recherche adopté: modèle booléen

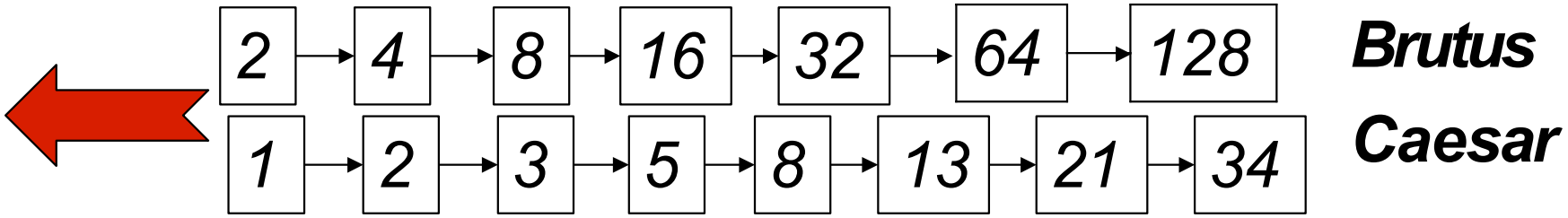
caesar

Term	N docs	Tot Freq	Ptr
ambitious	1	1	1
be	1	1	2
brutus	2	2	3
capitol	1	1	5
caesar	2	3	6
did	1	1	
enact	1	1	
hath	1	1	
I	1	2	
i'	1	1	
it	1	1	
julius	1	1	
killed	1	2	
let	1	1	
me	1	1	
noble	1	1	
so	1	1	
the	2	2	
told	1	1	
you	1	1	
was	2	2	
with	1	1	

Doc #	Freq
2	1
2	1
1	1
2	1
1	1
1	1
2	2
1	1
1	1
2	1
1	2
1	1
2	1
1	1
1	2
2	1
1	1
2	1
2	1
1	1
2	1
2	1
2	1
1	1
2	1
2	1

Répondre à une demande
Exemple requête avec plusieurs mots

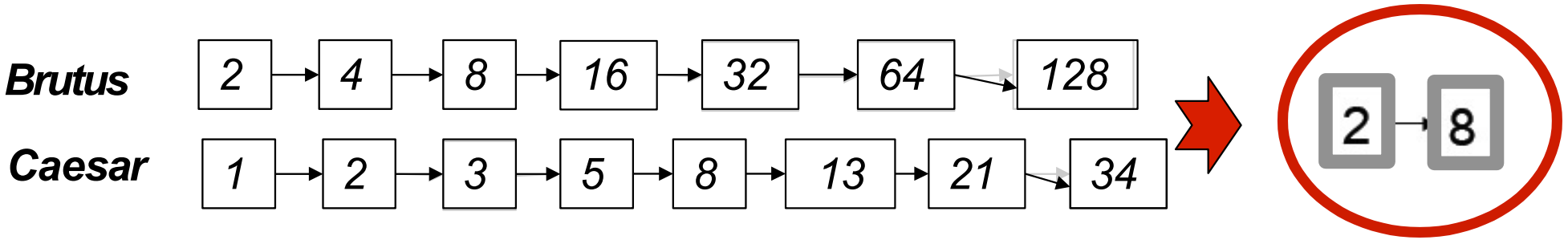
- Soit la requête :
  - ***Brutus*** *AND* ***Caesar***
  - Chercher ***Brutus*** dans le dictionnaire;
    - Sélectionner sa liste postings.
  - Chercher ***Caesar*** dans le dictionnaire ;
    - Sélectionner sa liste postings.
- – "Fusion" des deux postings:



Algorithme Fusion

- Parcourir les deux *postings* simultanément
- Les deux listes sont **ordonnées**
- Si les longueurs des listes sont x et y, l'algo est en O(x +y)

```
fusion = <>
id1 = l1[0], id2 = l2[0]
Tant que les listes ne sont pas vides
    si id1 = id2 alors
        ajouter(fusion, id1)
        id1 = suivant(l1)
        id2 = suivant(e2)
    sinon
        si id1 < id2 alors
            id1 = suivant(l1)
        sinon
            id2 = suivant(l2)
```



Limites de l'indexation textuelle classique

- Dans l'indexation textuelle classique: les mots sont souvent utilisés pour indexer des documents mais:
  - L'information sémantique n'est pas prise en compte
  - En pratique on s'intéresse à l'extraction des représentants de concepts
  - Un représentant d'un concept est la manifestation physique de ce concept

Limites de l'indexation textuelle classique

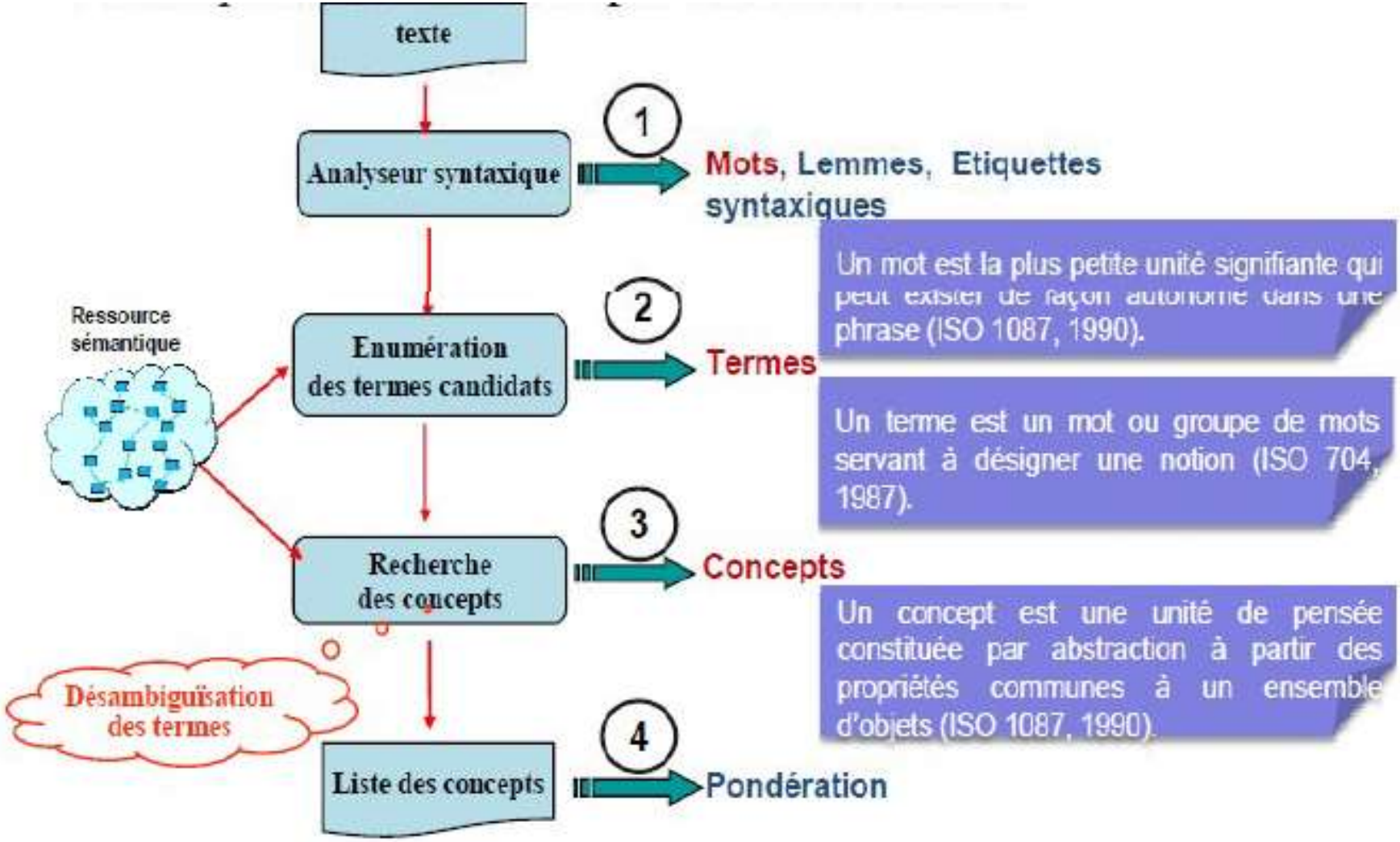
- Exemple
  - Un document **D** qui porte sur les bus qui mènent de Tunis à Siliana
  - Un utilisateur qui cherche un train pour aller de Tunis à Siliana

==> Un SRI classique ne permet pas de trouver D malgré que D est pertinent

  - L'information sémantique (la relation is-a)
    - Bus est un moyen de transport
    - Train est un moyen de transport

Indexation sémantique

- Principe: utiliser les concepts dans l'indexation



Travail à réaliser

- Préparer un exposé (.ppt) qui présente les techniques d'indexation textuelle sémantique
- Travail noté (+1 dans la note du DS)
- Travail facultatif

- Vous pouvez travailler en groupe (5 étudiants au maximun)
  - Chaque étudiant doit présenter une partie de l'exposé
  - Les exposés doivent être déposés en version ppt et envoyés par mail

Exercice 5

- Soient des descripteurs des 2 documents D1 et D2:
- D1 = {efficacité, recherche, mesurée, précision, moyenne}
- D2 = {modèles, recherche, efficaces, langage, vectoriel}

1. Calculer, pour chaque terme, TF (Term Frequency) :
   a. Utiliser la mesure simple
   b. Utiliser la mesure normalisée suivante:

$$TFi,j = \frac{f(Ti,Dj)}{1{,}5x\frac{longueur\_doc\_Dj}{longueur\_moy\_doc} + f(Tj,Dj) + 0{,}5}$$

$longueur\_doc\_Dj$ = nombre de descripteurs du document Dj

   c. Expliquer la différence entre les deux mesures

Exercice 5

2. On veut maintenant calculer le poids Wij associé à chaque terme. Pour IDF (Inverse Document Frequency), on considère la mesure du log.
   a. Donner la formule de IDF en expliquant la signification de chaque terme ;
   b. Donner le poids W de chaque terme (considérer la deuxième mesure de TF pour cette question);

3. Donner le fichier inversé correspondant.

Corrigé exercice 5 (1/4)

Longueur_doc_D1=5 Longueur_doc_D2=5
Longueur_moy_doc=10/2=5

	D1		D2	
Termes	TF	TF normalisée	TF	TF normalisée
efficacité	1	[1/(1.5+1+0.5)] = 0.33	0	0
recherche	1	1/3=0.33	1	1/3= 0.33
mesurée	1	1/3 = 0.33	0	0
précision	1	1/3= 0.33	0	0
moyenne	1	1/3= 0.33	0	0
modèles	0	0	1	1/3= 0.33
efficaces	0	0	1	1/3= 0.33
langage	0	0	1	1/3= 0.33
vectoriel	0	0	1	1/3= 0.33

Corrigé exercice 5 (2/4)

- La deuxième mesure permet de normaliser TF par rapport à la longueur du document.

- Formule de IDF

$$IDFi = \log_{10}(\frac{N}{Ni})$$

  - Ni : nombre de documents ou le terme Ti apparait
  - N est le nombre total de documents dans le corpus

Corrigé exercice 5 (3/4)

	D1	D2
Termes	**W = TF*IDF (D1)**	**W = TF*IDF (D2)**
efficacité	0.33*log(2) =0.1	0
recherche	0	0
mesurée	0.33*log(2) =0.1	0
précision	0.33*log(2) =0.1	0
moyenne	0.33*log(2) =0.1	0
modèles	0	0.33*log(2)=0.1
efficaces	0	0.33*log(2) =0.1
langage	0	0.33*log(2) =0.1
vectoriel	0	0.33*log(2) =0.1

Corrigé exercice 5 (4/4)
Fichier inversé

Dictionnaire

Terme	Nb doc	Total Freq	Ptr
efficaces	1	1	1
efficacité	1	1	2
langage	1	1	3
mesurée	1	1	4
modèles	1	1	5
moyenne	1	1	6
précision	1	1	7
recherche	2	2	8
vectoriel	1	1	**10**

Posting simple

Doc #	Freq
2	1
1	1
2	1
1	1
2	1
1	1
1	1
1	1
2	1
2	1