Recherche d'information

Akli ABBAS
abbasakli@gmail.com
Département Informatique
Université de Bouira

Plan du cours

- **Chapitre 1 :** Les notions de bases de la recherche d'information
- **Chapitre 2 :** Les modèles de recherche d'information
- **Chapitre 3 :** Les Stratégies de recherche
- **Chapitre 4 :** Evaluation des systèmes de recherche d'information

Qu'est ce que la RI ?

- **Recherche d'information** (RI) est une branche de l'informatique qui s'intéresse à l'**acquisition**, l'**organisation**, le **stockage**, la **recherche** et la **sélection d'information** «**salton1968**»

- Ensemble des **méthodes et techniques** pour l'acquisition, l'organisation, le stockage, la recherche et **la sélection d'information pertinente pour un utilisateur**

Bref historique de la RI

- **1940:** Apparition des SRI, focalisation de la RI sur les applications dans des bibliothèques.
- **1950:** Apparition du modèle booléen et l'élaboration de petites expérimentations sur des petites collections de documents.
- **1960 et 1970:** Apparition du système SMART (G. Salton, 1971), développement d'une **méthodologie d'évaluation de système et conception de corpus de test** pour évaluer des systèmes différents.
- **1980:** Développement de l'intelligence artificielle, ainsi on tentait d'intégrer des techniques de l'IA en RI (système expert).
- **1990 et 1995:** L'apparition d'internet, la RI a été modifié et sa problématique plus élargie (traitement des documents multimédia).

Domaine très visible

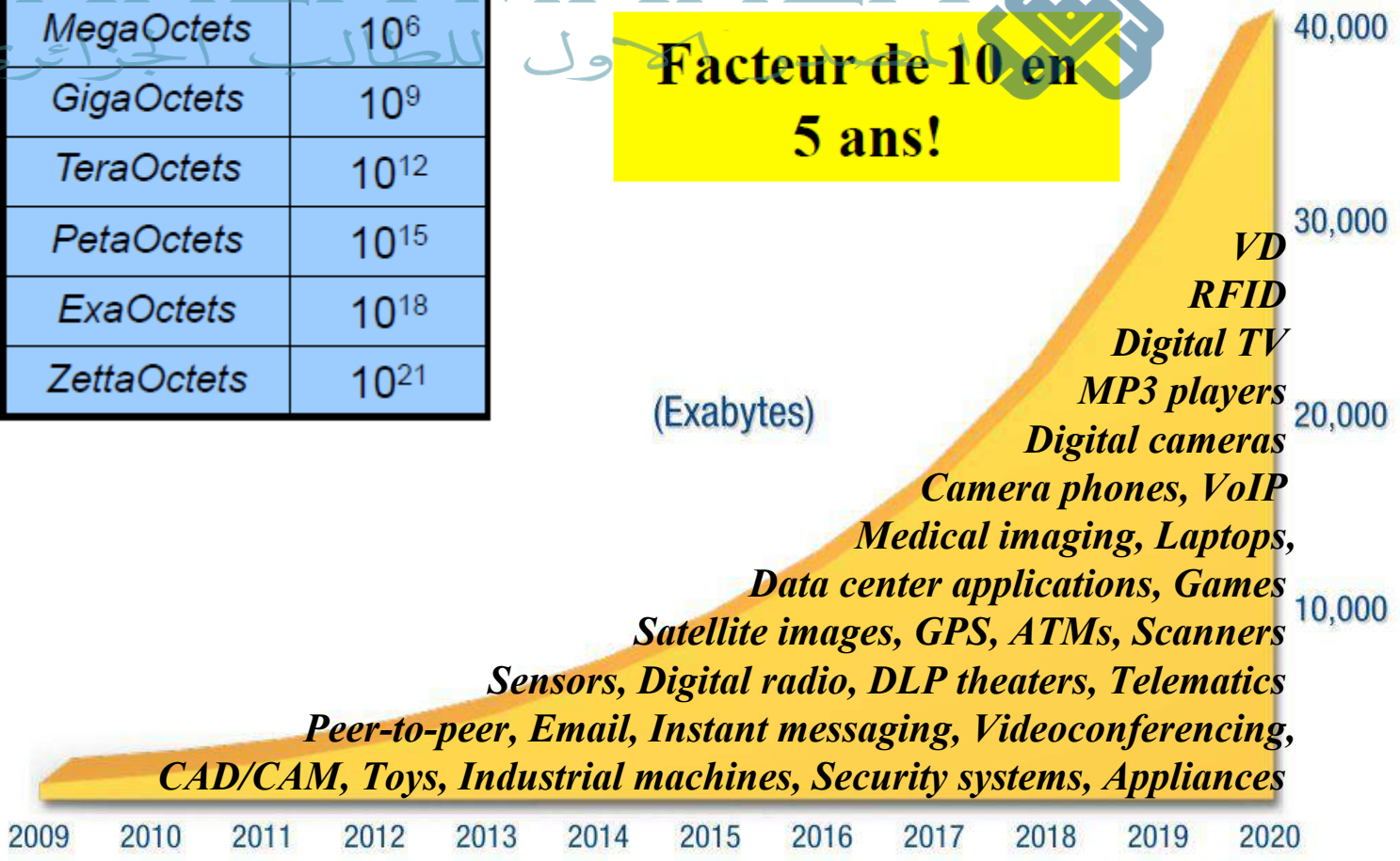
.. Mais pas seulement

- Plusieurs domaines d'application
  - **Internet (Web, Forum/Blog search, news)**
  - **Entreprises (entreprise search)**
  - **Bibliothèques numériques «digital library»**
  - **Domaine spécialisé (médecine, droit, littérature, chimie, mathématique, brevets, software, …)**
  - **Nos propres PC (Yahoo! Desktop search)**

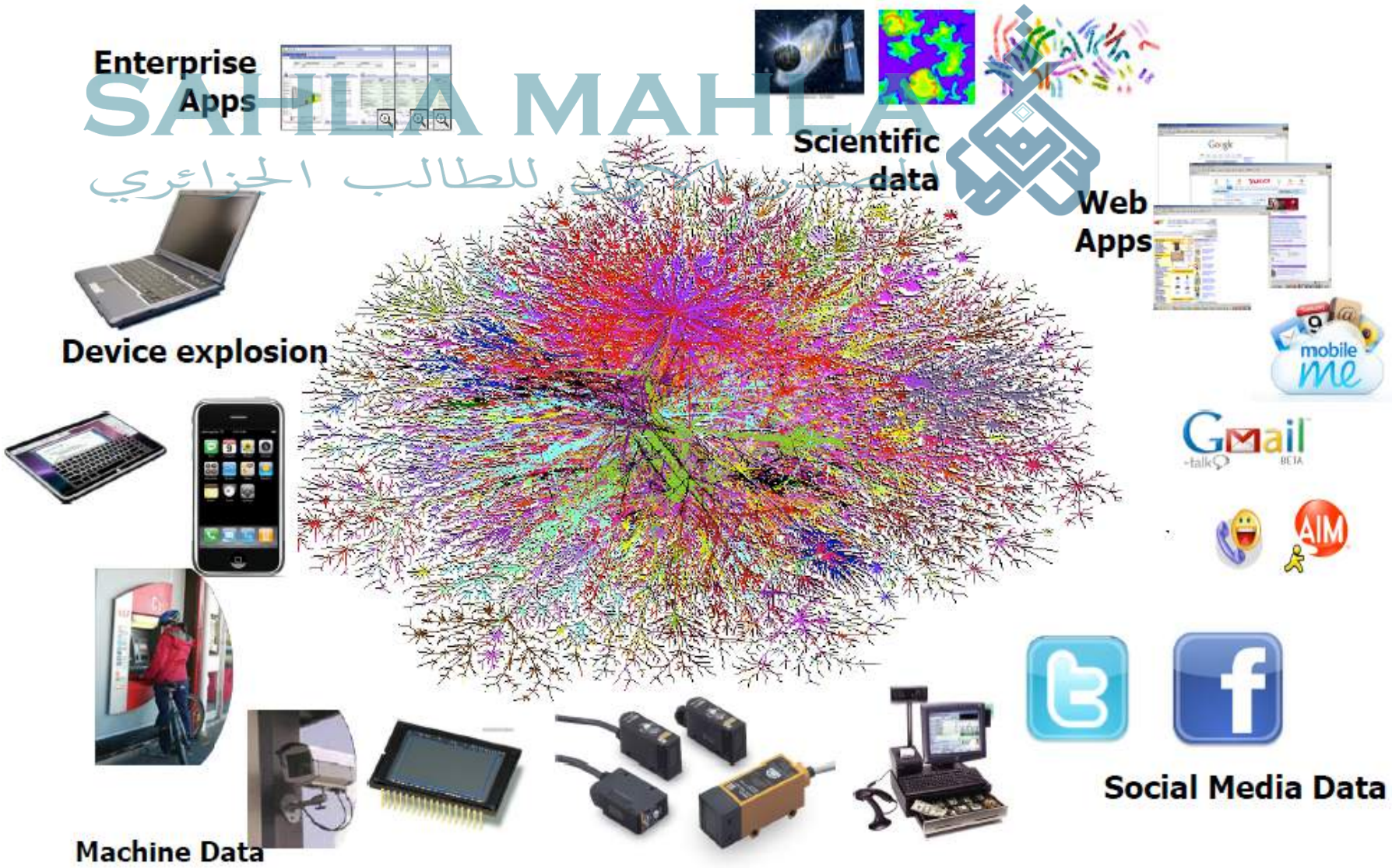
Gros volumes d'information

KiloOctets	10^3
MegaOctets	10^6
GigaOctets	10^9
TeraOctets	10^{12}
PetaOctets	10^{15}
ExaOctets	10^{18}
ZettaOctets	10^{21}

Facteur de 10 en 5 ans!



Origine

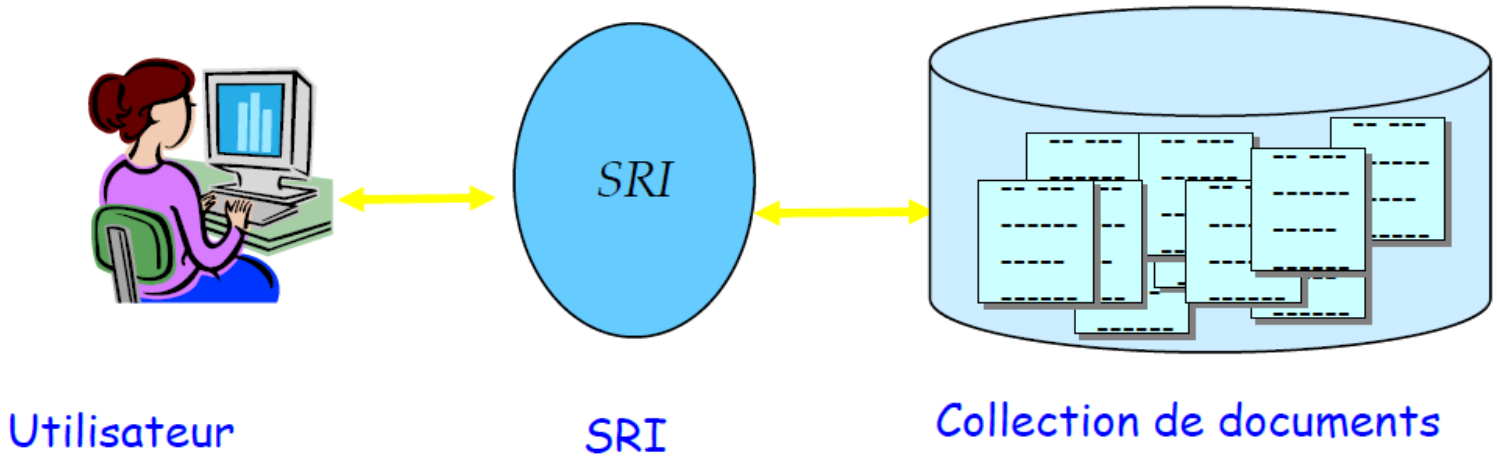


Origine

- **L'information (numérique) est disponible partout et avec un gros volumes.**
- Création **des systèmes de recherche d'information**.

Définition

- Un **Système de Recherche d'Information (SRI)** est un programme (ensemble de programmes) informatique qui a pour but de sélectionner des **informations pertinentes** répondant à des **besoins utilisateurs**



Donnée Vs. Information Vs. Connaissance

Système de gestion de Base de données

Système de Recherche d'information

Données :

Chaîne de caractères + valeurs associées à des objets, des personnes et des événements : (15)

Select .. From … where

Information :

Signification (explication/ description) des données, données intelligible (compréhensible):

(15° C - relevé à 18 h, sous abri, à Bouira)

Connaissance :

Information découverte, comprise et partagée par une communauté

(étant donné qu'on est à Bouira 15°C en février c'est plutôt froid)

Découverte de connaissance (information mining)

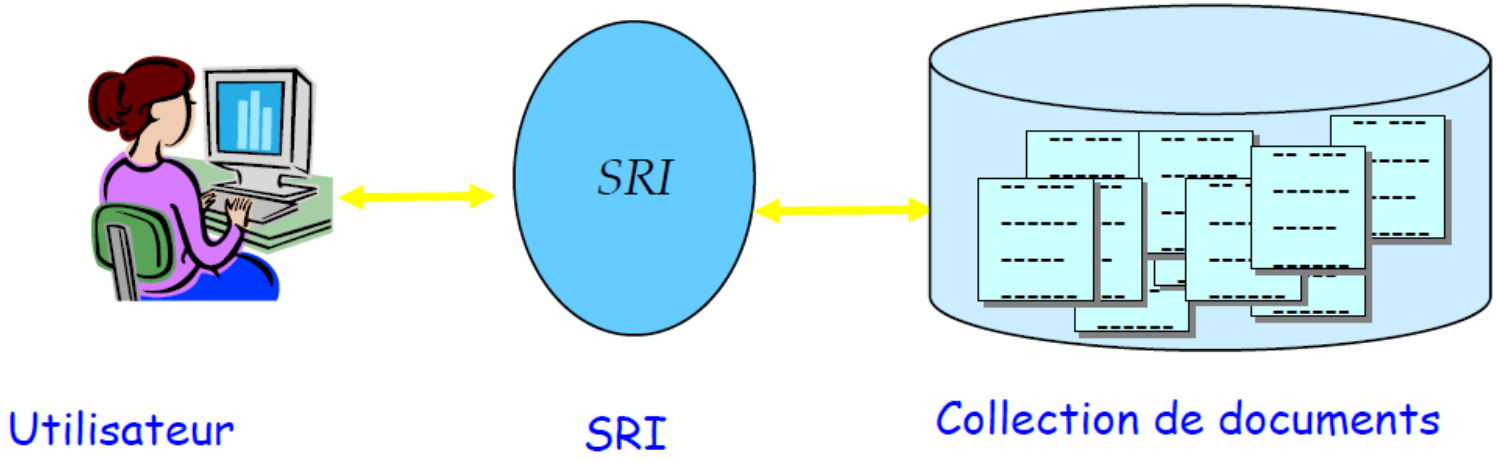
Information

- **Formes**
  - **Texte**
  - **images, sons, vidéo, graphiques, etc.**
- **Propriétés**
  - **Structure**

    Non structuré OU semi structuré (XML) (HTML)
  - **Hétérogénéité**
    - langage (multilingues)
    - media (multimédia)
    - structures

Définition d'un SRI

- Un Système de Recherche d'Information (SRI) est un programme (ensemble de programmes) informatique qui a pour but de sélectionner des **informations pertinentes** répondant à des **besoins utilisateurs**



Pertinence ?

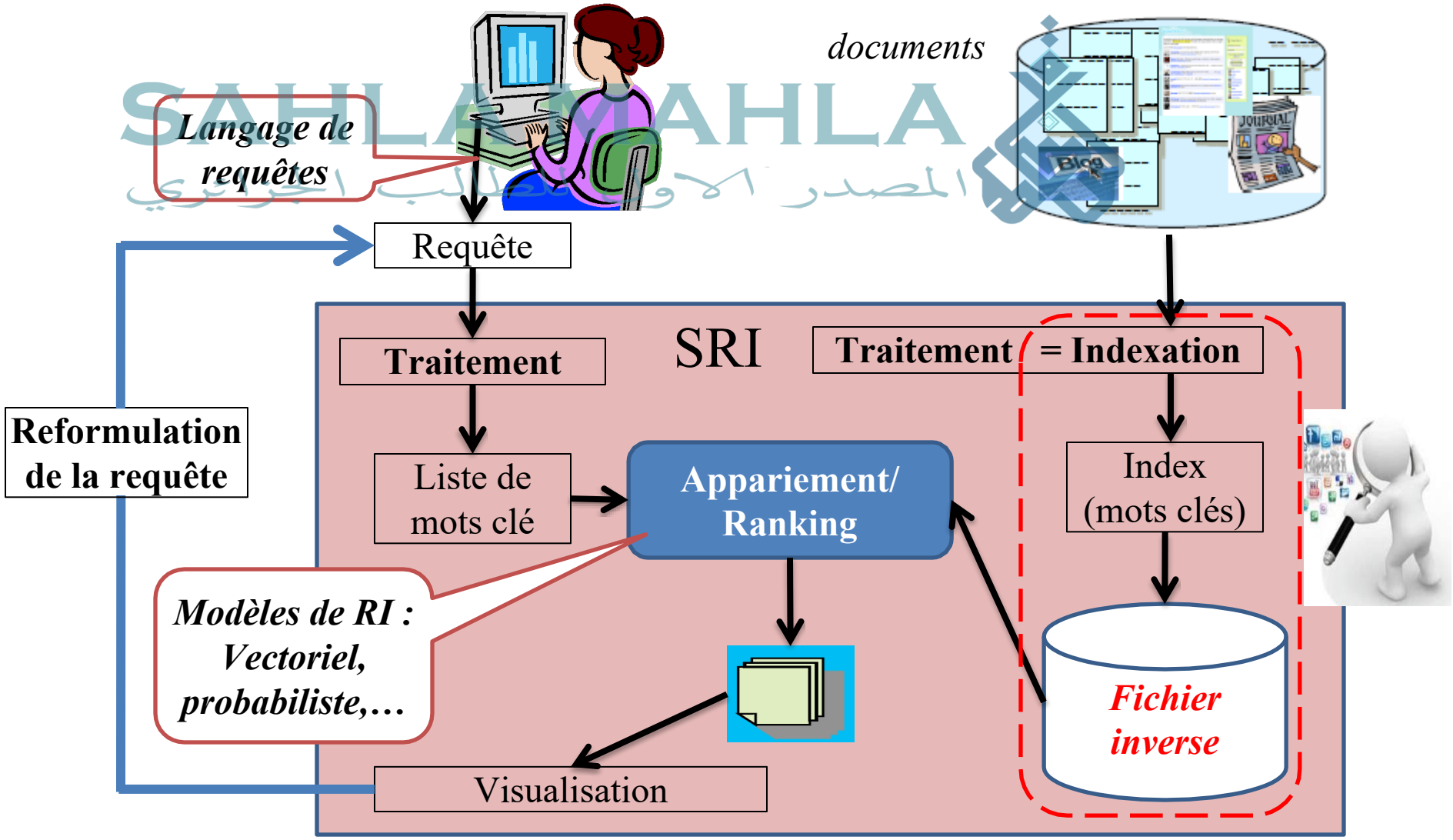
La notion de pertinence peut être appréhendée à deux niveaux :

➢**Niveau utilisateur :** la pertinence correspond à la satisfaction de l'utilisateur par apport à l'ensemble des documents restitués par le SRI. (**pertinence subjective, cognitive)**

➢**Niveau système :** le système mesure un degré de pertinence, une valeur de similitude entre un document et une requête. (**pertinence algorithmique, objective)**

Le but de tout SRI est de rapprocher la pertinence système de la pertinence utilisateur.

Processus de RI



Indexation ?

- Indexation = représentation de l'information

Def1: Consiste à créer un ensemble de mots clés reflétant aux mieux le **contenue sémantique** du document, cette liste de mots clés sera plus facilement exploitable lors du processus de la RI

Def2 : Processus permettant de construire un ensemble d'éléments « clés » permettant de caractériser le contenu d'un document / retrouver ce document en réponse à une requête

- **Éléments clés**
  - Information textuelle
    - mots simples : pomme
    - groupe de mots : pomme de terre
  - Image
    - Couleurs, formes

Indexation ?

- **Les approches d'indexation**
  - ✓ Manuelle (expert en indexation)
  - ✓ Automatique (ordinateur)
  - ✓ Semi-automatique (combinaison des deux)
- **Basée sur**
  - ✓ Un langage contrôlé (lexique/thesaurus/ontologie/réseau sémantique)
  - ✓ Un langage libre (éléments pris directement des documents)

Un langage contrôlé ?

- Lexique
  - Liste de mots clés
- Liste hiérarchique
  - de concepts
  - de notations (codes)
- Thésaurus
  - Liste de mots clés + relation sémantiques entre les mots clés
- Ontologie
  - Liste concepts + relations entre les concepts

Un langage contrôlé ?

- Liste hiérarchique (de concepts & de notations (codes))

A. Anatomy

B. Organisms

C. Diseases

C1. Bacterial infections

C2. Virus diseases

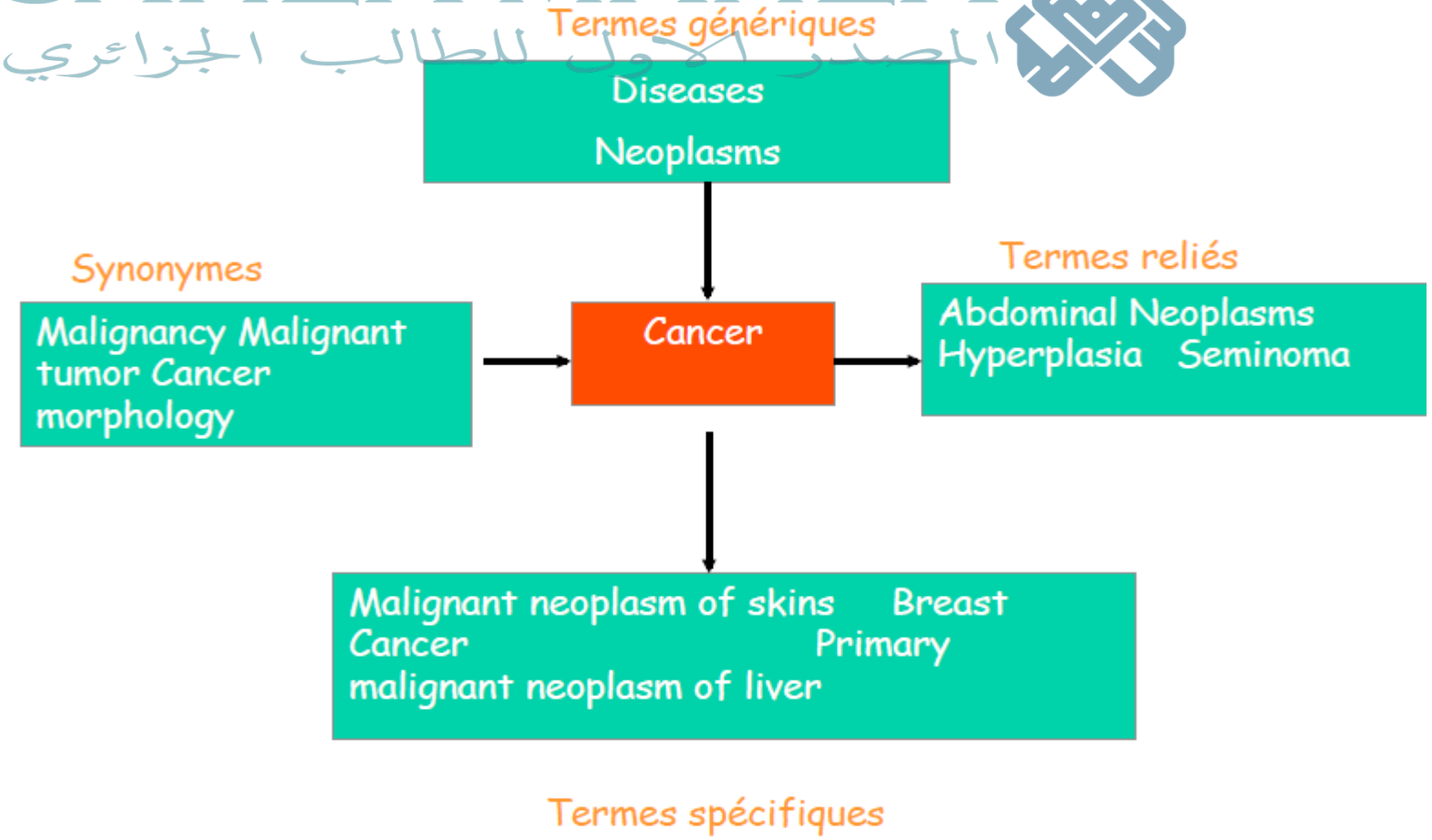
C 21. arbovirus infection

C 22. Encephalitis, Epidemic

C3. Parasitic diseases

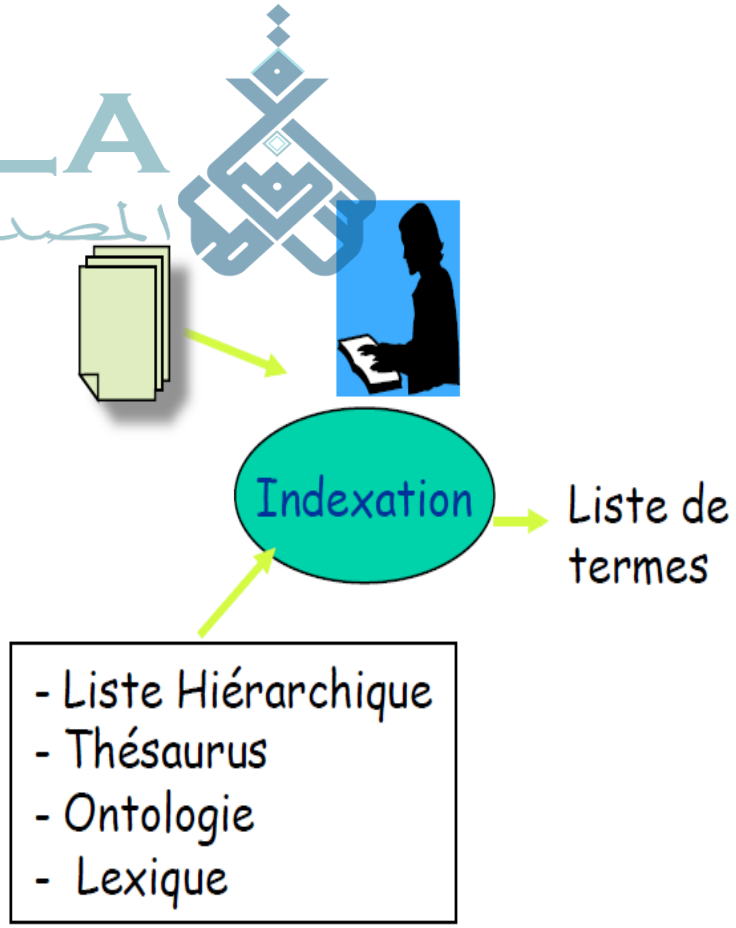
Un langage contrôlé ?

- Thésaurus : Liste de mots clés + relation sémantiques entre les mots clés



Indexation manuelle ?

- Choix des mots effectué par des indexeurs
- Basée sur un vocabulaire contrôlé
- Approche utilisée souvent dans les bibliothèques, les centres de documentation
- Dépend du savoir faire de l'indexeur



Indexation manuelle ?

Avantage du vocabulaire contrôle

- Permet la recherche par concepts (par sujets, par thèmes), plus intéressante que la recherche par mots simples
- Permet la classification (regroupement) de documents (par sujets, par thème)
- Fournit une terminologie standard pour indexer et rechercher les documents

Indexation manuelle ?

Inconvénients du vocabulaire contrôle

- **Indexation très coûteuse**
  – Pour construire le vocabulaire
  – Pour affecter les concepts (termes) aux documents (imaginer cette opération sur le web)
- **Difficile à maintenir**
  – La terminologie évolue, plusieurs termes sont rajoutés tous les jours
- **Processus humain donc subjectif**
  – Des termes différents peuvent être affectés à un même document par des indexeurs différents
- **Les utilisateurs ne connaissent pas forcément le vocabulaire utilisé par les indexeurs**

Indexation Automatique ?

C'est le SRI qui génère les indexes des documents.

Approches :

- **Statistique** (distribution des mots) et/ou **TALN** (compréhension du texte)

Approche courante est plutôt statistique avec des hypothèses simples :

- Redondance d'un mot marque son importance
- Cooccurrence des mots marque le sujet d'un document

4 étapes :

– Étape 1 : Extraction de mots simples

– Étape 2 : Normalisation des mots extraits

– Étape 3 : Statistique sur les occurrences

– Étape 4: Construction du fichier inverse et pondération des mots

Indexation Automatique ?

Etape 1 : Extraction des mots

1. Extraire les termes (tockenization)

terme = suite de caractères séparés par (blanc ou signe de ponctuation, caractères spéciaux,…), Nombres

- ❑ Ce sont les index utilisés lors de la recherche

2. Suppression des mots « vides » (stoplist / Commo Words removal)

Mots trop fréquents mais pas utiles

– Exemples :

- **Anglais : the, or, a, you, I, us, …**
- **Français : le , la de , des, je, tu, …**

Indexation Automatique ?

Etape 2 : Normalisation des mots extraits

➢ **«Lemmatisation» (radicalisation) / (stemming)**

– Processus morphologique permettant de regrouper les variantes d'un mot

• Ex1 : économie, économiquement, économiste,⇒ **économ**

• Ex2 (pour l'anglais) : retrieve, retrieving, retrieval, retrieved, retrieves ⇒ **retriev**

➢ **Utilisation de règles de transformations**

– règle de type : condition action :

Ex : si mot se termine par s supprimer la terminaison

– Technique utilisée principalement pour l'anglais :

L'algorithme le plus connu est : Porter

Indexation Automatique ?

Etape 2 : Normalisation des mots extraits

- **Analyse grammaticale**
  - Utilisation de lexique (dictionnaire)
  - Tree-tagger (gratuit sur le net)
- **Troncature :** Tronquer les mots à X caractères
  - Tronquer plutôt les suffixes
  - Exemple troncature à 7 caractères : économiquement : écomoni

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

Indexation Automatique ?

Etape 3 : Statistique sur les occurrences

Pour chaque mot, on doit faire la statistique de sa fréquence d'occurrence dans le document. Ainsi, à chaque nouvelle occurrence d'un mot, on ajoute 1 dans sa fréquence.

Indexation Automatique ?

(Exemple)

✓ Texte : un système de recherche d ’informations (document) (SRI, base de données documentaires) permet d ’analyser, d ’indexer et de retrouver les documents pertinents répondant à un besoin d ’un utilisateur.

Etape 1 : Extraire les termes et suppression des mots vides

✓ système, recherche, informations, document, SRI, base, données, documentaires, analyser, indexer, retrouver, documents, pertinents, répondant, besoin , utilisateur

Etape 2 : Normalisation des mots extraits (troncature à 7)

✓ systeme, recherc, informa, documen, sri, base, donnee, documen, analyse, indexer, retrouv, documen, pertine, reponda, besoin, utilisa

Etape 3 : Statistique sur les occurrences

✓ systeme 1,recherc 1, informa 1, documen 3, sri 1, base 1, donnee 1, analyse1, indexer 1, retrouv 1, pertine 1, reponda 1, besoin 1, utilisa *1*

Indexation Automatique ?

Etape 4 : Construction du fichier inverse et pondération des mots

Une fois les documents indexés le résultat est que chaque document aura donc un descripteur / une représentation :

- Un descripteur :
  - Liste de mots
  - Fréquence de chaque mot

- Ces mots sont ensuite stockés dans une structure appelée **fichier inverse**

Indexation Automatique ?

Etape 4 : Construction du fichier inverse et pondération des 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	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

Indexation Automatique ?

Etape 4 : Construction du fichier inverse et pondération des 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	N docs	Tot Freq
ambitious	1	1
be	1	1
brutus	2	2
capitol	1	1
caesar	2	3
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

Indexation Automatique ?

Etape 4 : pondération des mots

Comment caractériser les termes importants dans un document ou plusieurs docuements?

- ……Pondération des termes

- Idée :
  - Les termes importants doivent avoir un poids fort

Approches de pondération

- **Plusieurs approches :**
  - **Tf, IDF** (approche plus répandue)
  - Pourvoir discriminatoire d'un terme
  - -….
- **Dépend aussi du modèle de RI.**

TF * IDF

➢**TF** : (*term frequency)* plus un terme est fréquent dans un document plus il est important dans la description de ce document

Exemple de *tf* :

$$\text{TF} = \begin{cases} freq(t,d) \\ 1+\log(freq(t,d)) \\ \dfrac{freq(t,d)}{\max_{\forall t' \in d}(t',d)} \\ \dfrac{freq(t,d)}{\sum_{\forall t' \in d} freq(t',d)} \end{cases}$$

Robertson TF : TF = TF/(TF+k) est souvent appelé“**Okapi TF**”

•K introduit pour tenir compte de la longueur des documents

$$tf = \frac{fréq.}{fréq. + 0.5 + 1.5 * \dfrac{longueur\_doc}{longueur\_moy\_doc}}$$

TF * IDF

➢ **IDF :** (Inverse Document Frequency) la fréquence du terme dans la collection (ensemble des documents).

✓ Désigne le pouvoir de discrimination d'un terme c.-à-d. qu'un terme distingue bien un document des autres documents.

$$idf(t) = \begin{cases} \log\left(\frac{N}{n_t}\right) \\ \log\left(\frac{N - n_t}{n_t}\right) \end{cases}$$

avec

N : le nombre de documents de la collection,

n_t : le nombre de documents contenant le terme t

➢**Le poids du terme dans un document**

$$w(t, d) = tf * idf$$

TD N°1

DOCUMENTS:

D1 : La mesure R-précision est pertinente pour la mesure de la précision moyenne
D2: Les modèles de recherche les plus efficaces sont le modèle de langage et le modèle vectoriel
D3: L'efficacité de la recherche est mesurée par la précision moyenne
D4 : Apparition du modèle booléen et l'élaboration de petites expérimentations sur des petites collections de documents

QUESTIONS :

- Donner la table des fréquences : terme, document, terme dans la collection
- Calculer TF*IDF de chaque terme
- Soit la requête Q: «Modèle de recherche efficace »
  - Calculer le degré de correspondance $\mathbf{R(D_i, Q)} = \sum w(t_q, D_i)$ représentant la somme des fréquences des termes de la requête t_q dans le document D_i.
  - Quel est le document qui sera classé en haut lors de la réponse.

Qualité de l'indexation

- **Exhaustive*(rappel)* :** Représente le nombre de documents pertinents extraits par rapport au nombre de documents pertinents (limiter le silence)

$$rappel = \frac{\text{Nombre de documents retrournés et pertinents}}{\text{Nombre de documents pertinents}}$$

- **Spécificité*(précision)* :** Représente le nombre de documents pertinents extraits par rapport au nombre de documents extraits. (Exactitude et précision des index, limiter le bruit)

$$précision = \frac{\text{Nombre de documents retrournés et pertinents}}{\text{Nombre de documents extraits}}$$

Les modèles de RI

Qu'est ce qu'un modèle de RI ?

➢Un modèle est une abstraction d'un processus (ici recherche d'info)

➢Les modèles de RI peuvent décrire

– Le processus de mesure de pertinence : comment les documents sont sélectionnés et triés

– L'utilisateur : besoin en information, interaction

– L'information

➢Les modèles de RI manipulent plusieurs variables :

les besoins, les documents, les termes, les jugements de pertinence , les utilisateurs, …

➢Les modèles de RI se distinguent par le principe d'appariement (matching) :

appariement exact /approché (Exact /Best matching

Appariement exact /Appariement approché

➢ Appariement exact

✓ Requête spécifie de manière précise les critères recherchés

✓ L'ensemble des documents respectant exactement la requête sont sélectionnés, mais pas ordonné.

➢ Appariement approché

✓ Requête décrit les critères recherchés dans un document

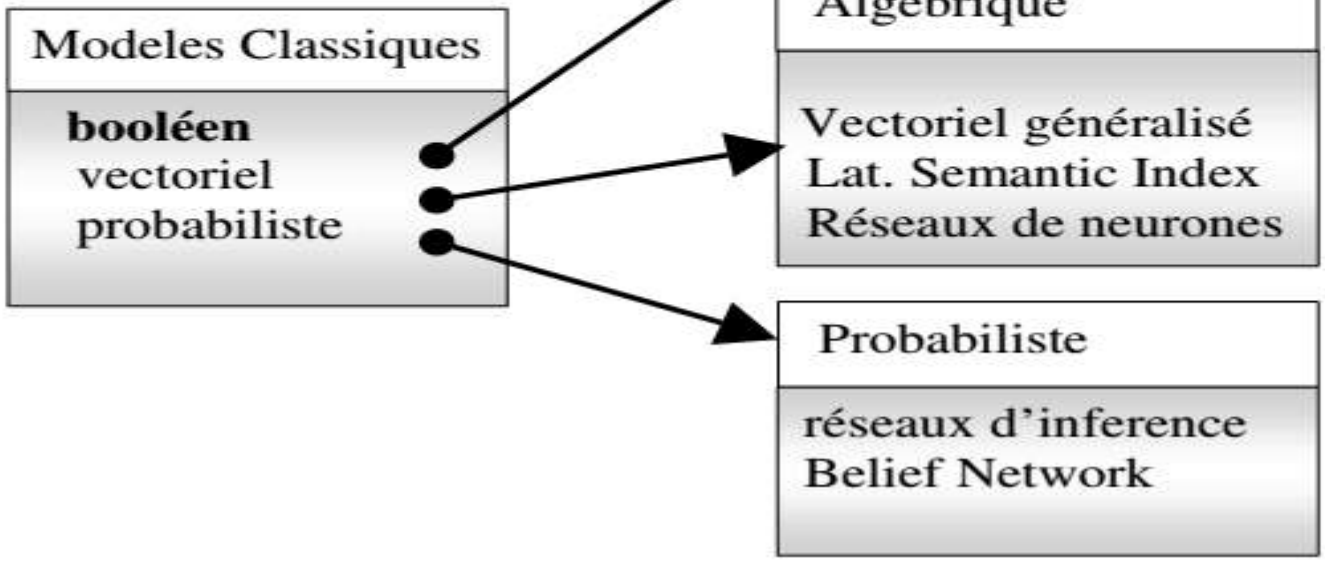
✓ Les documents sont sélectionnés selon un degré de pertinence (similarité/ probabilité) vis-à-vis de la requête et sont ordonné

Modèles de RI

Les différent modèles de la RI

- ✓ Modèle booléen (±1950)
- ✓Modèle vectoriel (±1970)
- ✓Modèle probabiliste (±1976)
- ✓Modèle connexionniste (réseaux de neurones)(±1989)
- ✓Modèle d'inférence (réseau d'inférence bayésien) (±1992)
- ✓Modèle LSI (Latent Sementic Indexing) (± 1994)
- ✓Modèle de langage (±1998)

Modèles de RI



Le Modèle Booléen

- Le premier modèle de RI
- Basé sur la théorie des ensembles
- Un document est représenté un ensemble de termes: $d = t_1 \wedge t_2 \wedge \ldots \wedge t_n$

–Ex : $d1(t_1,t_2,t_5)$; $d2(t_1,t_3,t_5,t_6)$; $d3(t_1,t_2,t_3,t_4,t_5)$

- Une requête est un ensemble de mots avec des opérateurs booléens :

AND ($\wedge$), OR($\vee$), NOT ($\neg$)

–Ex: $q = t_1 \wedge (t_2 \vee \neg t_3)$

- Appariement Exact basé sur la présence ou l'absence des termes de la requête dans les documents

- Appariement (d, q) = R (d, q)=1 ou 0

Le Modèle Booléen

La correspondance R(d, q) entre une requête et un document est déterminée de la façon suivante:

$R(d, t_i) = 1$ si $t_i \in d$; 0 sinon.

$R(d, q_1 \wedge q_2) = 1$ si $R(d, q_1) = 1$ et $R(d, q_2) = 1$; 0 sinon.

$R(d, q_1 \vee q_2) = 1$ si $R(d, q_1) = 1$ ou $R(d, q_2) = 1$; 0 sinon.

$R(d, \neg q_1) = 1$ si $R(d, q_1) = 0$; 0 sinon.

Le Modèle Booléen

Exemples :

Requête

$q = t_1 \wedge (t_2 \vee \neg t_3)$

Documents

$d1(t_1,t_2,t_5)$; $d2(t_1,t_3,t_5,t_6)$; $d3(t_1,t_2,t_3,t_4,t_5)$

Calculer la correspondance :

R(d1, q) = ?

R(d2, q) = ?

R(d3, q) = ?

Le Modèle Booléen

Les avantages

1. Ce modèle est simple à mettre en œuvre
2. la clarté conceptuelle des systèmes booléens

Les inconvénients

1. Tous les termes dans un document ou dans une requête étant pondérés de la même façon simple (0 ou 1) c'est à dire, indexation binaire
2. La sélection d'un document est basée sur une décision binaire
3. Pas d'ordre pour les documents sélectionnés
4. Formulation de la requête difficile pas toujours évidente pour beaucoup l'utilisateurs
5. Problème de collections volumineuses : le nombre de documents retournés peut être considérable

Modèle Vectoriel (Vector Space Model)

- Proposé par Salton dans le système SMART (Salton, G. 1970)

Idée de base :

Représenter les documents et les requêtes sous forme de vecteurs dans l'espace vectoriel engendré par tous les termes de la collection de documents

Un document Doc_i est représenté par un vecteur de dimension m :

$$Doc_i = (w_{i1}, w_{i2}, \ldots, w_{im}) \text{ pour } i = 1, 2, \ldots, n.$$

où w_{ij} est le poids (TF*IDF) du terme t_j dans le document Doc_i

n est le nombre de documents dans la collection,

m est le nombre de termes dans les documents de la collection.

Une requête qk est représentée par un vecteur dans le même espace des termes.

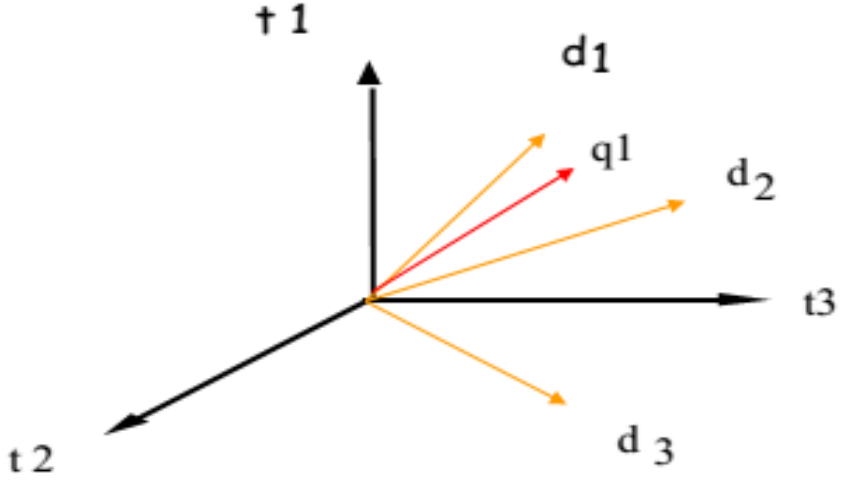
$q_k = (w_{k1}, w_{k2}, \ldots, w_{km})$. où w_{kj} est le poids de terme t_j dans la requête q_k.

Modèle Vectoriel (Vector Space Model)

Soit T<t_1, t_2, …, t_M> : ensemble des M termes de la collection

$D_i = (w_{i1}, w_{i2}, \ldots, w_{im})$

$q_k = (w_{k1}, w_{k2}, \ldots, w_{km})$



Modèle Vectoriel (Vector Space Model)

Une collection de **n** documents et **M** termes distincts peut être représentée sous forme de matrice

	T_1	T_2	....	T_M
D_1	w_{11}	w_{21}	...	w_{M1}
D_2	w_{12}	w_{22}	...	w_{M2}
:	:	:		:
:	:	:		:
D_n	w_{1n}	w_{2n}	...	w_{Mn}

La requête est également représentée par un vecteur.

Modèle Vectoriel (Vector Space Model)

Exemple :

– T (document, web, information, recherche, image, contenu) : ensemble des termes d'indexation

d1(document 2,web 1) ; d2(information 1, document 3, contenu 2)

q1 (image, web); q2(recherche, documentaire)

– Représentation vectorielle

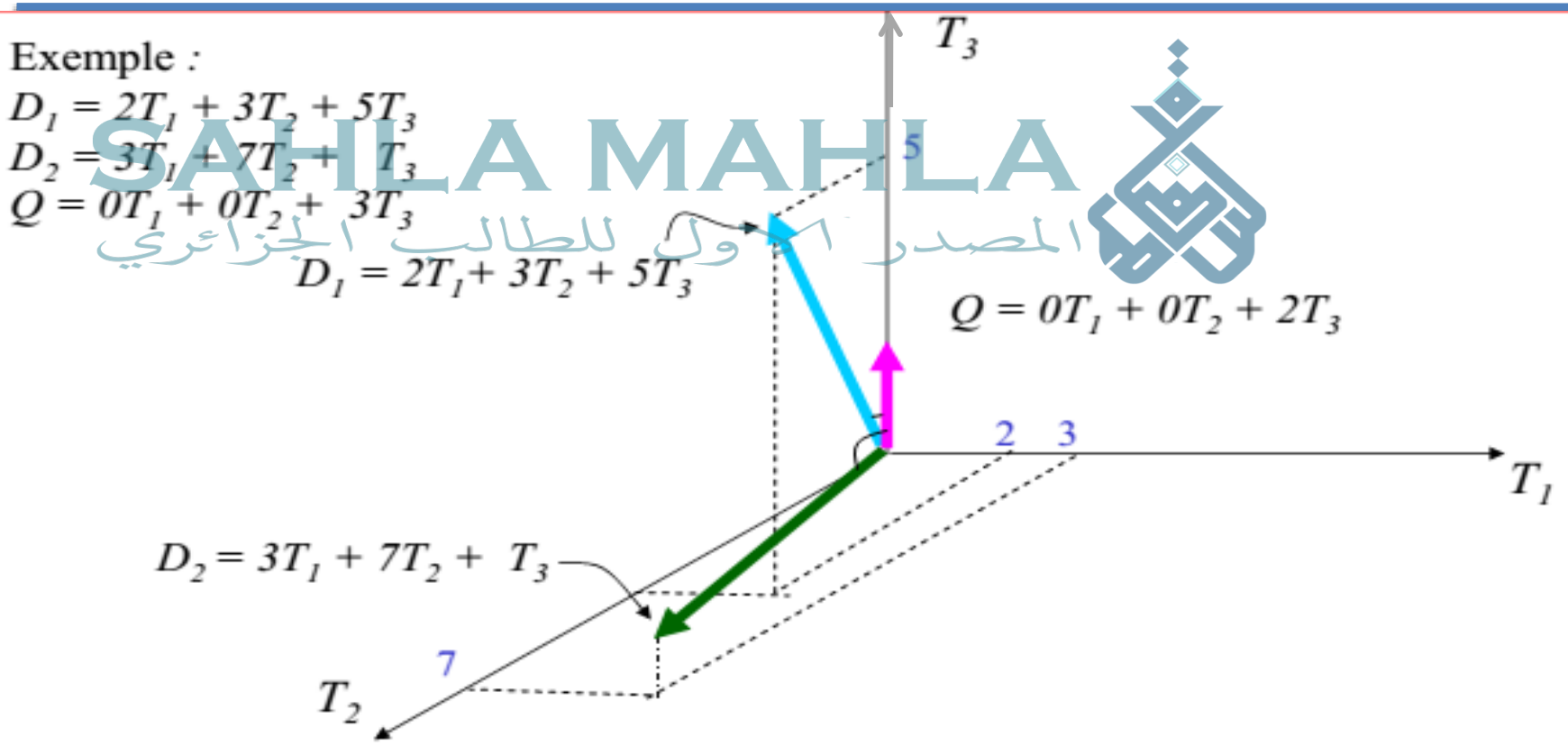
d1 (2,1,0,0,0,0)

d2 (?)

q1 (?)

q2 (?)

Modèle Vectoriel (Vector Space Model)



La pertinence est traduite en une similarité vectorielle :

un document est d'autant plus pertinent à une requête que le vecteur associé est similaire à celui de la requête.

Modèle Vectoriel (Vector Space Model)

Le degré de correspondance R(d, q):

Produit scalaire des deux vecteurs :

$$R(d_i, q_k) = \sum_{j=1}^{m} w_{ij} * w_{kj} \quad \text{pour } i = 1, 2, \ldots, n.$$

Cosinus de l'angle :

$$R(d_i, q_k) = Cos(\vec{d_i}, \vec{q_k}) = \frac{\sum_{j=1}^{m} w_{ij} * w_{kj}}{\sqrt{\sum_{j=1}^{m} w_{kj}^2 * \sum_{j=1}^{m} w_{ij}^2}} \quad \text{pour } i = 1, 2, \ldots, n.$$

Modèle Vectoriel (Vector Space Model)

Le degré de correspondance R(d, q):

Inner product	$\|X \cap Y\|$	$\sum x_i * y_i$
Coef. de Dice	$\frac{2*\|X \cap Y\|}{\|X\|+\|Y\|}$	$\frac{2*\sum x_i * y_i}{\sum x_i^2 + \sum y_j^2}$
Mesure du cosinus	$\frac{\|X \cap Y\|}{\sqrt{\|X\|} * \sqrt{\|Y\|}}$	$\frac{\sum x_i * y_i}{\sqrt{\sum x_i^2 * \sum y_j^2}}$
Mesure du Jaccard	$\frac{\|X \cap Y\|}{\|X\|+\|Y\|-\|X \cap Y\|}$	$\frac{\sum x_i * y_i}{\sum x_i^2 + \sum y_j^2 - \sum x_i * y_i}$

Avantages

L'un des avantages du modèle vectoriel réside dans sa simplicité conceptuelle et de mise en œuvre.

Il offre aussi des moyens simples pour classer les résultats d'une recherche

Il est robuste et performant dans les tests.

Inconvénients

Approche vectorielle considère chaque terme comme étant indépendant des autres (pas de liens entre termes).

Extension du modèle Booléen

Extension du modèle Booléen

Prendre en compte l'importante des termes dans les documents et/ou dans la requête

- Possibilité d'ordonner les documents sélectionnés
- Comment étendre le modèle booléen ?

– Interpréter les conjonctions et les disjonction

- **Deux modèles :**

– Modèle flou- fuzzy based model (basé sur la logique floue)

– Modèle booléen étendu- extended boolean model

Modèle booléen étendu

Prendre en compte l'importante des termes dans les documents et/ou dans la requête

- Possibilité d'ordonner les documents sélectionnés
- Comment étendre le modèle booléen ?
  - Interpréter les conjonctions et les disjonction
- **Deux modèles :**
  - Modèle flou- fuzzy based model (basé sur la logique floue)
  - Modèle booléen étendu- extended boolean model

Modèle booléen étendu

Combinaison des modèles booléen et vectoriel

–Document : liste de termes pondérés

–Requête booléenne

–Utilisation des distances algébriques pour mesurer la pertinence d'un document vis-à-vis à d'une requête

Modèle booléen étendu - appariement

- Considérons
  - **dj= (w_{1j}, w_{2j} ,… ,w_{tj})**
  - **q : requête à deux termes (t_1 , t_2)**

$$R(d_j, t_1 \vee t_2) = \sqrt{\frac{w_{1j}^2 + w_{2j}^2}{2}}$$

$$R(d_j, t_1 \wedge t_2) = 1 - \sqrt{\frac{(1 - w_{1j})^2 + (1 - w_{2j})^2}{2}}$$

Modèle booléen étendu - appariement

Exemple :

Documents	Booléen				booléen étendu	
	A	B	A ou B	A et B	A ou B	A et B
D1	1	1	1	1	?	?
D2	1	0	1	0	?	?
D3	0	1	1	0	?	?
D4	0	0	0	0	?	?

Modèle booléen étendu - appariement

Exemple :

Documents	Booléen				booléen étendu	
	A	B	A ou B	A et B	A ou B	A et B
D1	1	1	1	1	**1**	**1**
D2	1	0	1	0	**1/sqr(2)**	**1-1/sqr(2)**
D3	0	1	1	0	**1/sqr(2)**	**1-1/sqr(2)**
D4	0	0	0	0	**0**	**0**

Modèle booléen étendu - appariement

- Généralisation
  - Distance euclidienne à plusieurs dimensions
  - Utilisation de la p-norm
- Considérons

- $d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$

- q : requête composée de m termes ($t_1, t_2, \ldots, t_m$)

$$R(d_j, q\,or) = \left(\frac{w_{1j}^p + w_{2j}^p + \ldots + w_{mj}^p}{m}\right)^{\frac{1}{p}}$$

$$R(d_j, qand) = 1 - \frac{\left((1-w_{1j})^p + (1-w_{2j})^p + \ldots + (1-w_{mj})^p\right)^{\frac{1}{p}}}{m^{\frac{1}{p}}}$$

$$R(d_j, qnot) = 1 - R(d_j, q)$$

Modèle booléen étendu - appariement

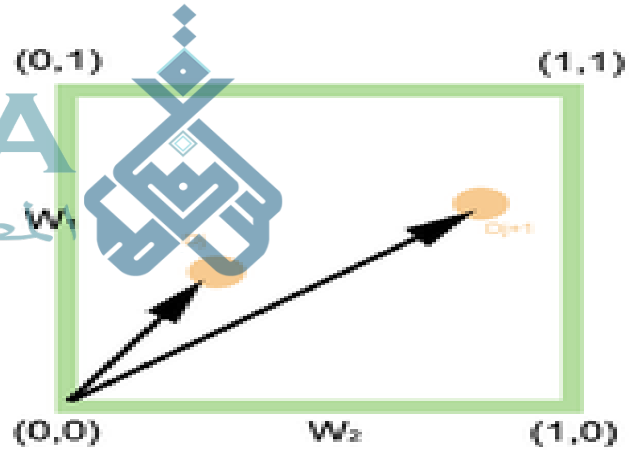
- Généralisation

- $d_j = (w_{1j}, w_{2j}, \ldots, w_{tj})$

- Si la requête et les documents sont pondérés $q(q_1, q_2, \ldots, q_m)$

$$R(d_j, qor) = \left(\frac{\sum q_i^p * w_{1j}^p}{\sum q_i^p}\right)^{\frac{1}{p}}$$

$$R(d_j, qand) = 1 - \left(\frac{\sum q_i^p * (1 - w_{1j})^p}{\sum q_i^p}\right)^{\frac{1}{p}}$$



Similarité d'une requête de type ET (**t_1 et t_2**) entre une requête Q et les documents D_j et D_{j+1}

$$R(d_j, t_1 \wedge t_2) = 1 - \sqrt{\frac{(1-w_{1j})^2 + (1-w_{2j})^2}{2}}$$

Similarité d'une requête de type OU (**t_1 ou t_2**) entre une requête Q et les documents D_j et D_{j+1}

$$R(d_j, t_1 \vee t_2) = \sqrt{\frac{w_{1j}^2 + w_{2j}^2}{2}}$$

TESTE

D1 « L'objectif d'un système de recherche d'informations est de répondre à une requête d'un utilisateur »

D2 « L'information traitée dans ce domaine de recherche est l' explication ou la description d'une donnée»

Q1 (recherche, information)

Questions :

1- On suppose que les termes d'indexation sont obtenus par extraction des **mots simples** (avec élimination des mots vides) mais sans troncature. **Précisez l'ensemble des termes obtenus.**

2- Proposez une fonction de pondération et justifiez votre choix

3- Calculez degré de correspondance $\mathbf{R(D_i, Q)} = \sum w(t_q, D_i)$ (similarité) entre la requête et les documents et dites quel document sera restitué au premier

Modèle booléen basé sur des ensembles flous

Un document est représenté comme un ensemble de termes pondérés comme suit:

$$d = \{\ldots, (t_i, a_i), \ldots\}$$

t_i est le terme, ai est le poids associer au terme t_i .

Le degré de correspondance (évaluation d' une requête) :

Évaluation 1: [Zadeh]

$$R(d, t_i) = a_i$$

$$R(d, q_1 \wedge q_2) = \min(R(d, q_1), R(d, q_2)).$$

$$R(d, q_1 \vee q_2) = \max(R(d, q_1), R(d, q_2)).$$

$$R(d, \neg q_1) = 1 - R(d, q_1).$$

Modèle booléen basé sur des ensembles flous

Un document est représenté comme un ensemble de termes pondérés comme suit:

$$d = \{\ldots, (t_i, a_i), \ldots\}$$

t_i est le terme, ai est le poids associer au terme t_i .

Le degré de correspondance (évaluation d'une requête) :

Évaluation 2: [Lukaswicz]

$$R(d, t_i) = a_i$$

$$R(d, q_1 \wedge q_2) = R(d, q_1) * R(d, q_2).$$

$$R(d, q_1 \vee q_2) = R(d, q_1) + R(d, q_2) - R(d, q_1) * R(d, q_2).$$

$$R(d, \neg q_1) = 1 - R(d, q_1).$$

Evaluation de performance dans les SRI

Objectif:

Evaluer la performance d'une approche, d'une technique, d'un système

–En RI, on ne mesure pas la performance absolue d'un système / technique / approche car non significative

–Mais, ..

- Evaluation comparative entre approches
- Mesurer la performance relative de A par rapport à B

Critères d'évaluation

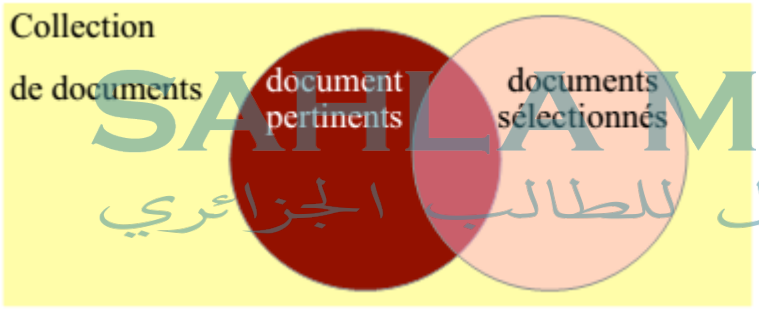
- Identifier la tâche à évaluer
- Identifier les critères (Cleverdon 66)
  - acilité d'utilisation du système
  - oût accès/stockage
  - résentation des résultats
  - **apacité d'un système à sélectionner des documents pertinents.**

Deux facteurs

- **Rappel**
  - La capacité d'un système à sélectionner tous les documents pertinents de la collection

- **Précision**
  - La capacité d'un système à sélectionner que des documents pertinents

Evaluation de performance dans les SRI



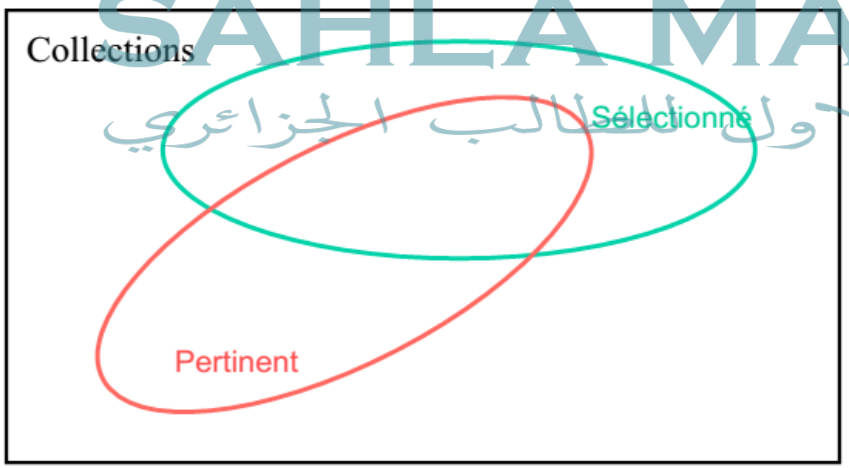
	Sélectionnés	Non Sélectionnés
Non pertinents	Non Pertinents & Sélectionnés	Non Pertinents & Non Sélectionnés
Pertinents	Pertinents & Sélectionnés	Pertinents & Non Sélectionnés

$$rappel = \frac{Nombre\ de\ documents\ pertinents\ séléctionnés}{Nombre\ total\ de\ documents\ pertinents}$$

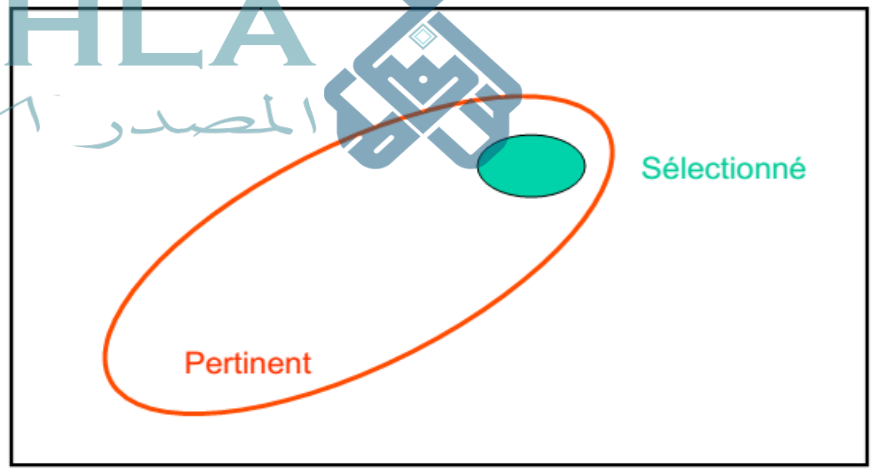
$$précision = \frac{Nombre\ de\ documents\ pertinents\ \ sélectionnés}{Nombre\ total\ de\ documents\ sélectionnés}$$

Evaluation de performance dans les SRI

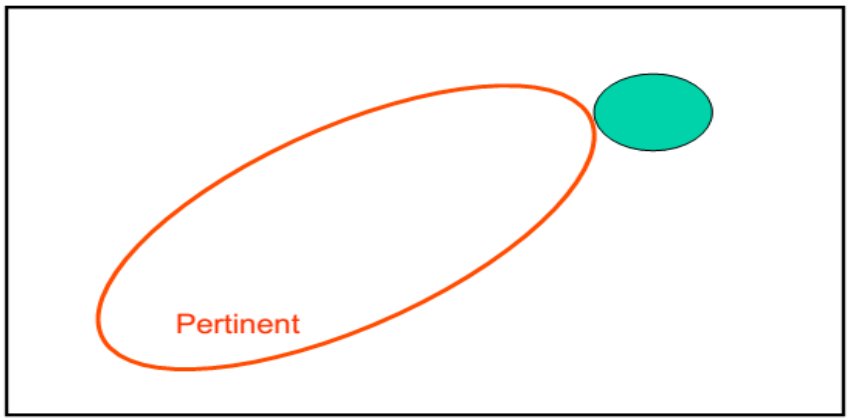
Collections



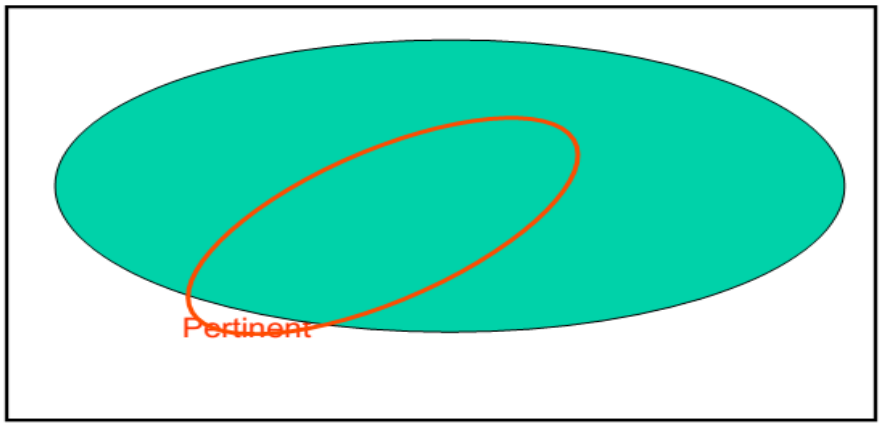
Précision très élevée, rappel très faible



Précision très faible, rappel très faible (en fait, 0)

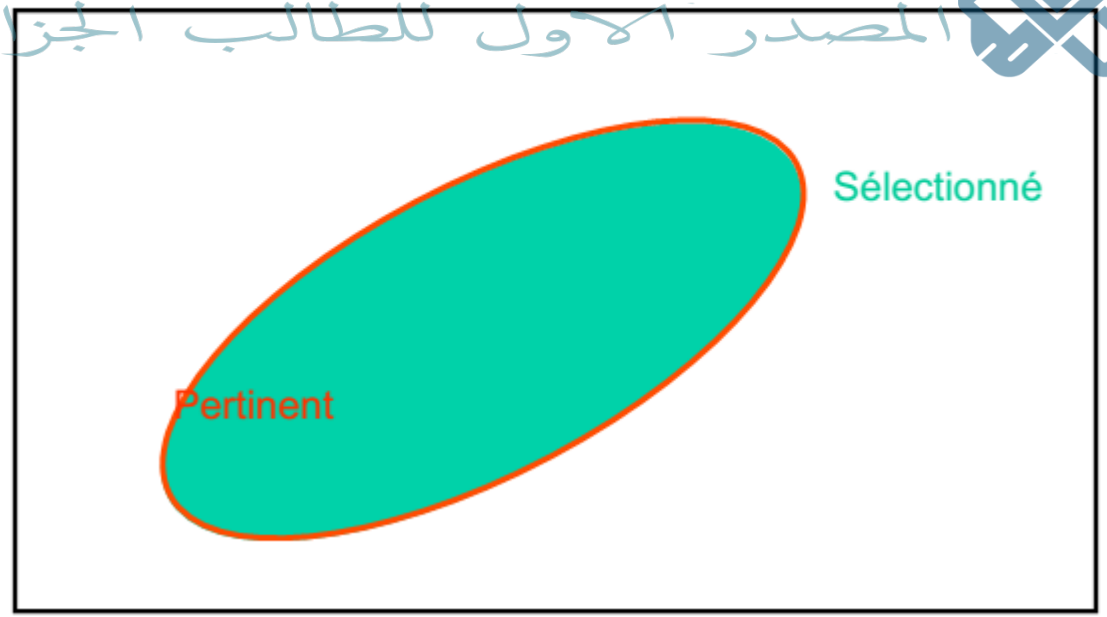


Rappel élevé, mais précision faible

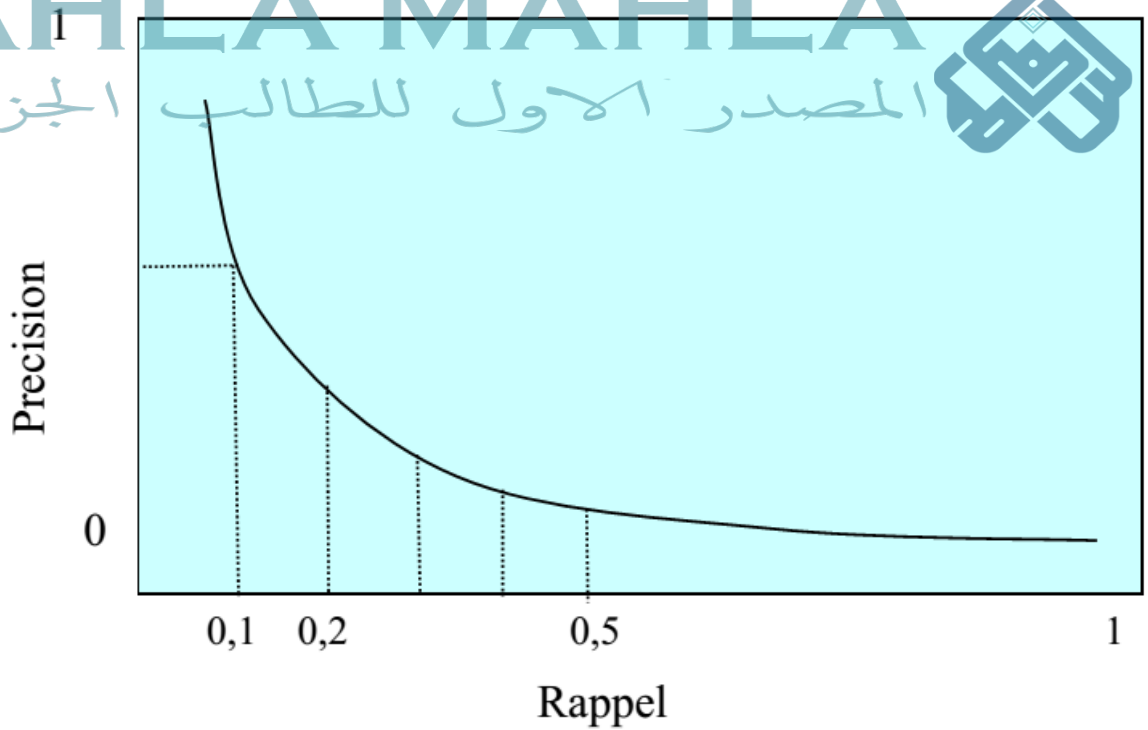


Evaluation de performance dans les SRI

Précision élevée, rappel élevé (idéal, mais difficile)



Lien entre Rappel et Précision



Précision moyenne : une seule valeur reliant le rappel et précision

Démarche d'évaluation

• **Démarche Analytique (formelle):**

– Difficile pour les SRI, car plusieurs facteurs : pertinence, distribution des termes, etc. sont difficiles à formaliser mathématiquement.

• **Démarche Expérimentale**

– par « benchmarking ».

– Evaluation effectuée sur des collections de tests

– Collection de test : un ensemble de documents, un ensemble de requêtes et des pertinences (réponses positives pour chaque requêtes)

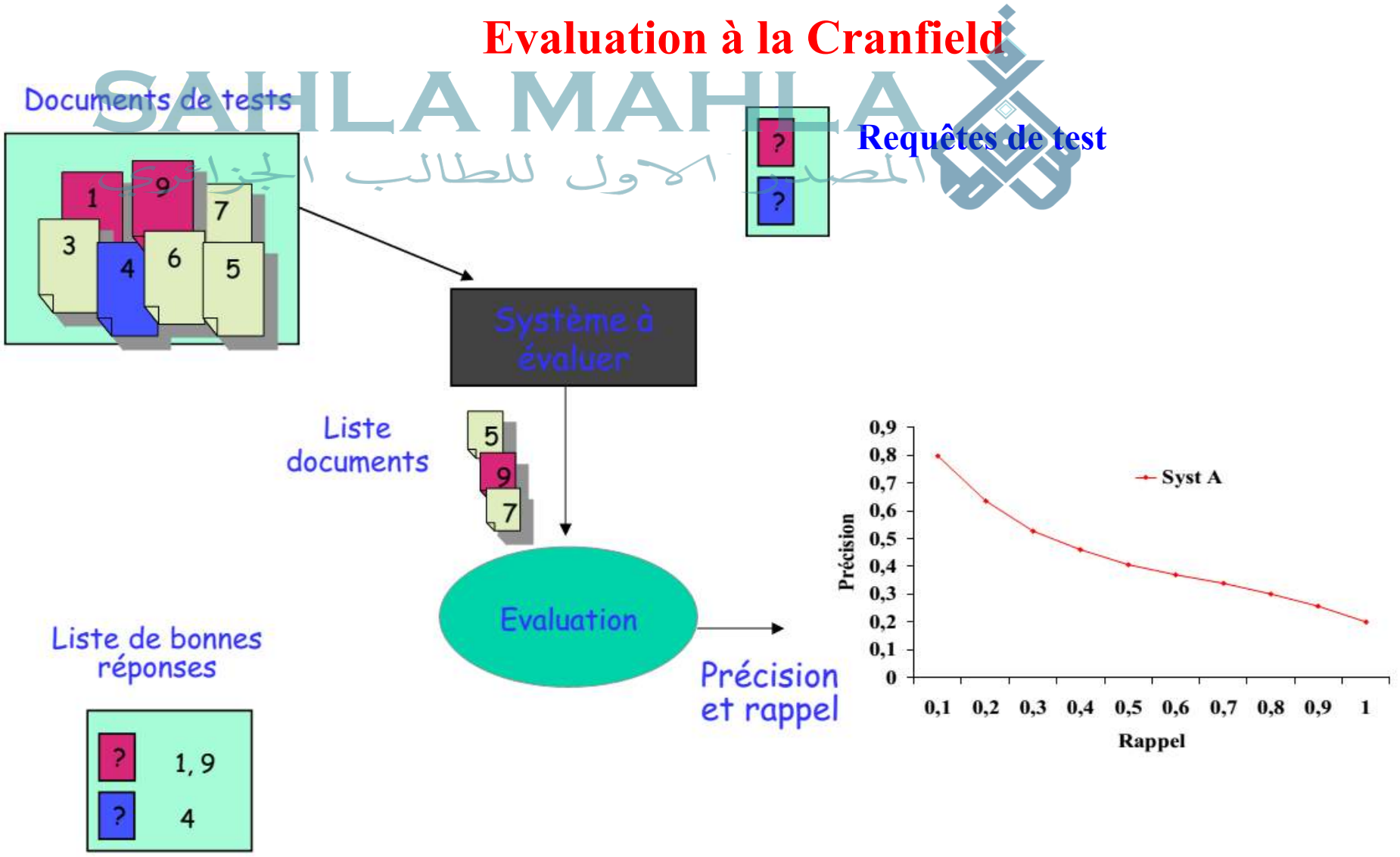
Démarche expérimentale

- **Lancée dès les années 1960, par Cleverdon, dans le cadre du projet Cranfield**

- **Objectif du projet Cranfield**
  - Construire des collections de test
  - Evaluer les systèmes sur ces collections de test

Evaluation de performance dans les SRI

Evaluation à la Cranfield



Calcul du rappel et de la précision

- On suppose qu'on dispose d'une collection de tests

–Lancer chaque requête sur la collection de tests.

–Marquer les documents pertinents par rapport à la liste de test.

–Calculer le rappel et la précision à pour chaque document pertinent de la liste.

Calcul du rappel et de la précision- Exemple

n	doc #	relevant
1	588	x
2	589	x
3	576	
4	590	x
5	986	
6	592	x
7	984	
8	988	
9	578	
10	985	
11	103	
12	591	
13	772	x
14	990	

Le nombre total de documents pertinents est = 6

R=1/6=0.167; P=1/1=1

R=2/6=0.333; P=2/2=1

R=3/6=0.5; P=3/4=0.75

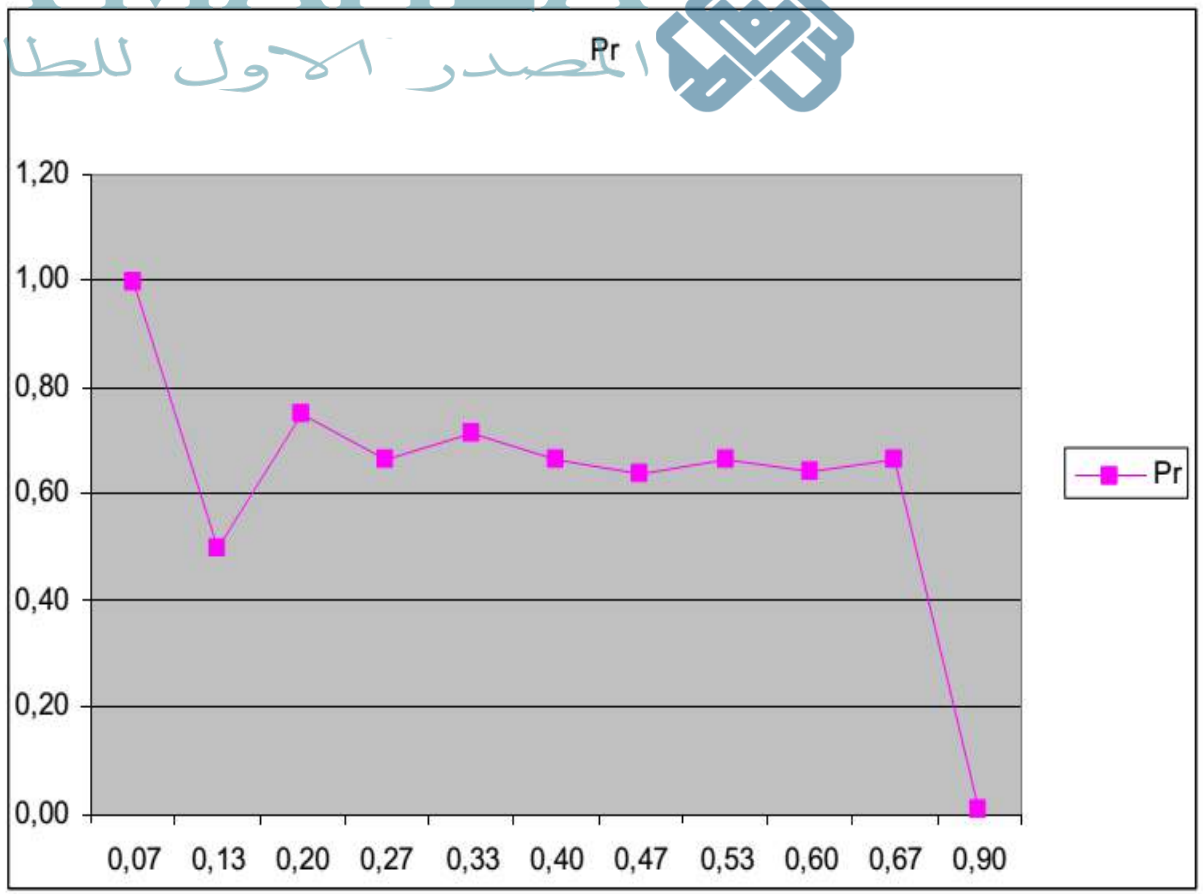
R=4/6=0.667; P=4/6=0.667

R=5/6=0.833; p=5/13=0.38

Il manque un document pertinent. On atteindra pas le 100% de rappel

Calcul du rappel et de la précision- Exemple 2

Ra	Pr
0,07	1,00
0,13	0,50
0,20	0,75
0,27	0,67
0,33	0,71
0,40	0,67
0,47	0,64
0,53	0,67
0,60	0,64
0,67	0,67
0,90	0,01



Interpolation de la courbe Rappel/Précision

• Interpoler une précision pour chaque point de rappel :

– $r_j \in \{0.0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$

– $r_0 = 0.0, r_1 = 0.1, \ldots, r_{10} = 1.0$

• La précision interpolée au point de rappel r_j est égale à la valeur maximale des précisions obtenues aux points de rappel r, tel que r >= r_j

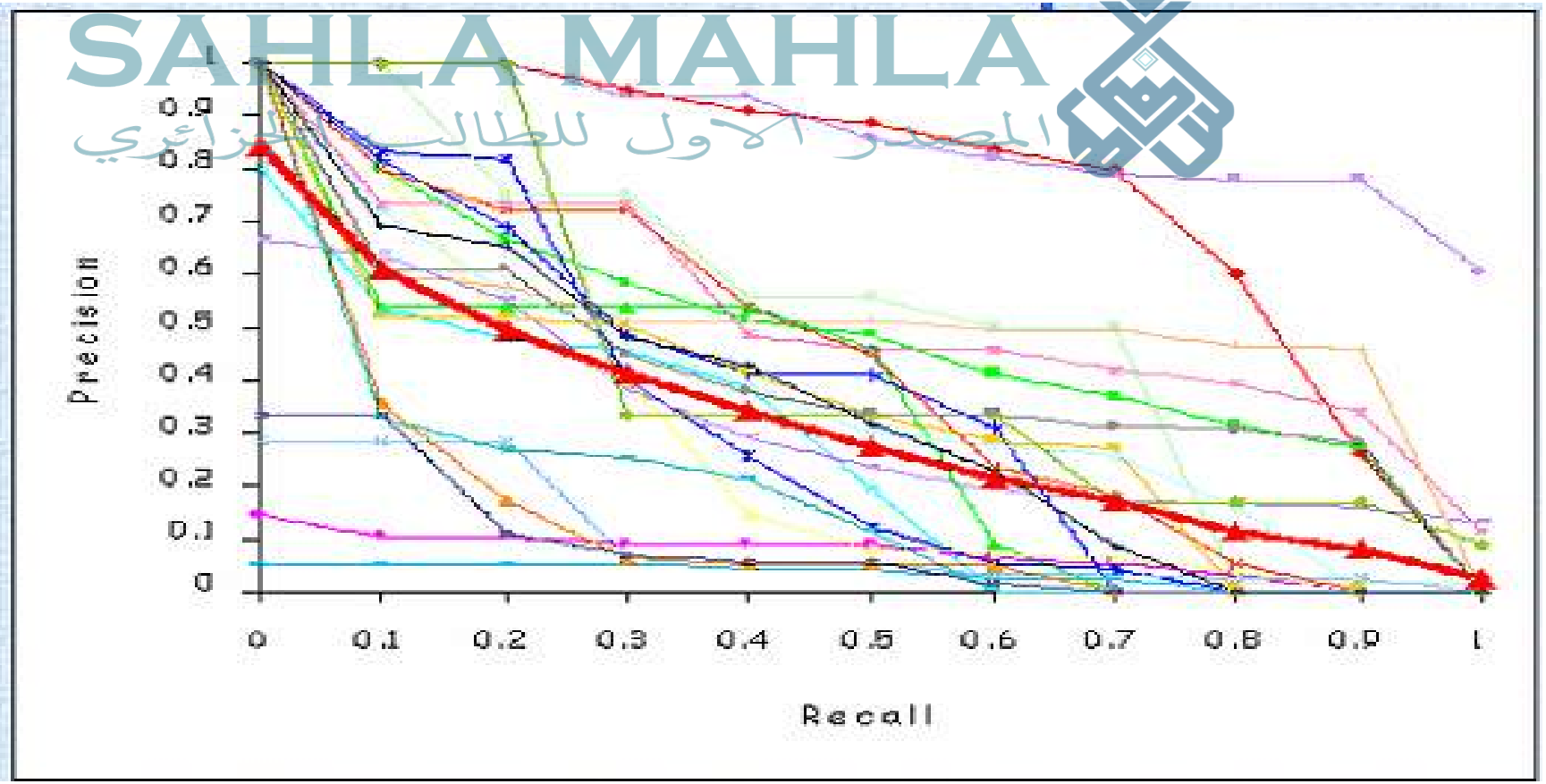
$$P(r_j) = \max_{r \geq r_j} P(r)$$

Exemple Interpolation des Précisions

Ra	Pr
0,07	1,00
0,13	0,50
0,20	0,75
0,27	0,67
0,33	0,71
0,40	0,67
0,47	0,64
0,53	0,67
0,60	0,64
0,67	0,67
0,90	0,01

Ra	Pr
0,0	1
0,1	0.75
0,2	0.75
0,3	0.71
0,4	0.67
0,5	0.67
0,6	0.67
0,7	0.01
0,8	0.01
0,9	0.01
1	0

R-P courbes sur l'ensemble des requêtes



Illisible, difficile de comparer deux approches/systèmes requête par requête
On a besoin d'une moyenne entre les requêtes

Courbe des moyennes sur plusieurs requêtes

Macro moyenne

–Calculer la précision moyenne à chaque point de rappel pour l'ensemble des requêtes.

–Tracer la courbe rappel-précision

Evaluation de performance dans les SRI

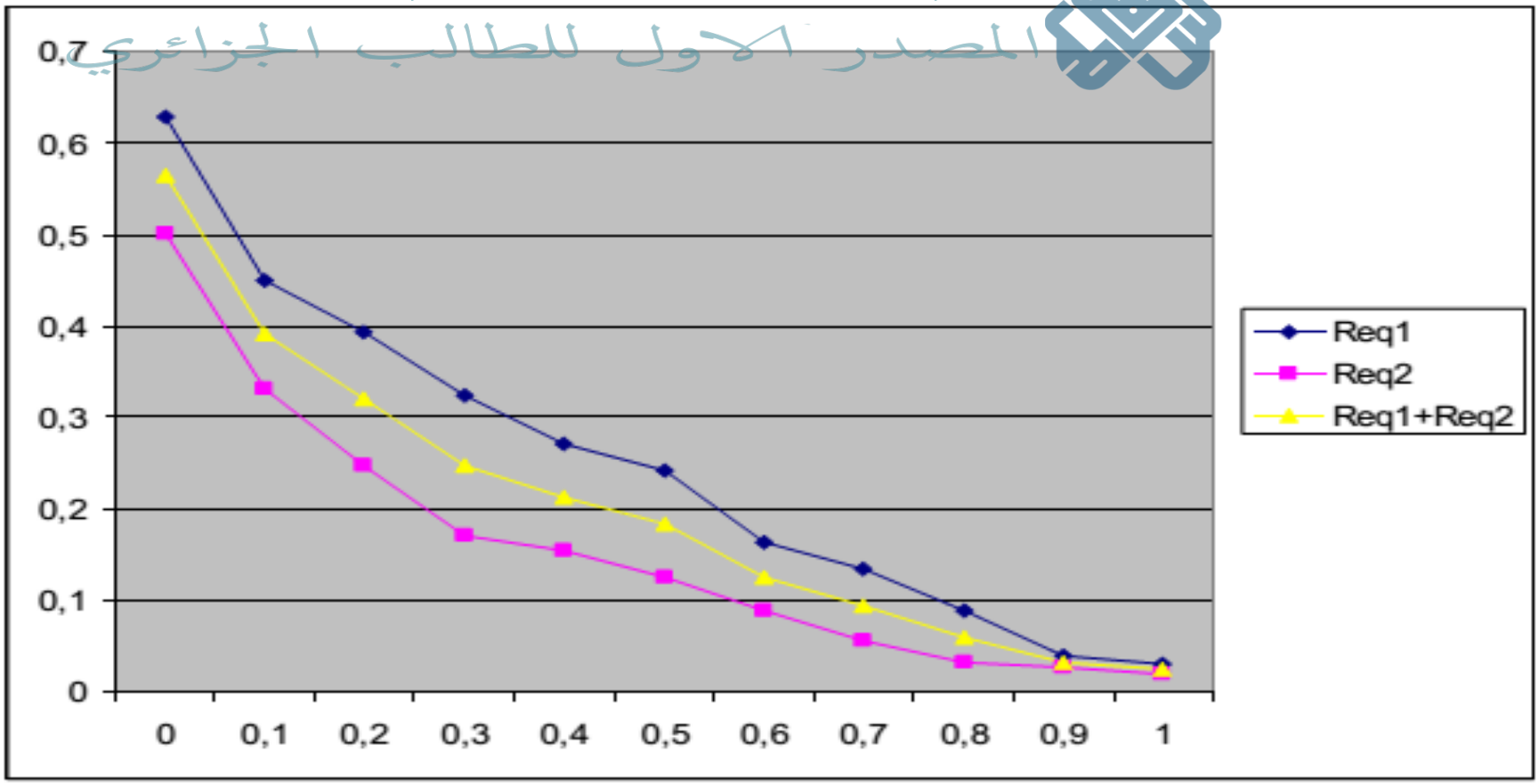
Exemple

Requete1	
R	Pr
0	0,629
0,1	0,451
0,2	0,393
0,3	0,3243
0,4	0,271
0,5	0,2424
0,6	0,164
0,7	0,134
0,8	0,09
0,9	0,04
1	0,031
AvrPrec	0,2329

Requete2	
R	Pr
0	0,5017
0,1	0,332
0,2	0,248
0,3	0,171
0,4	0,155
0,5	0,125
0,6	0,089
0,7	0,056
0,8	0,032
0,9	0,027
1	0,02
AvrPrec	0,1443

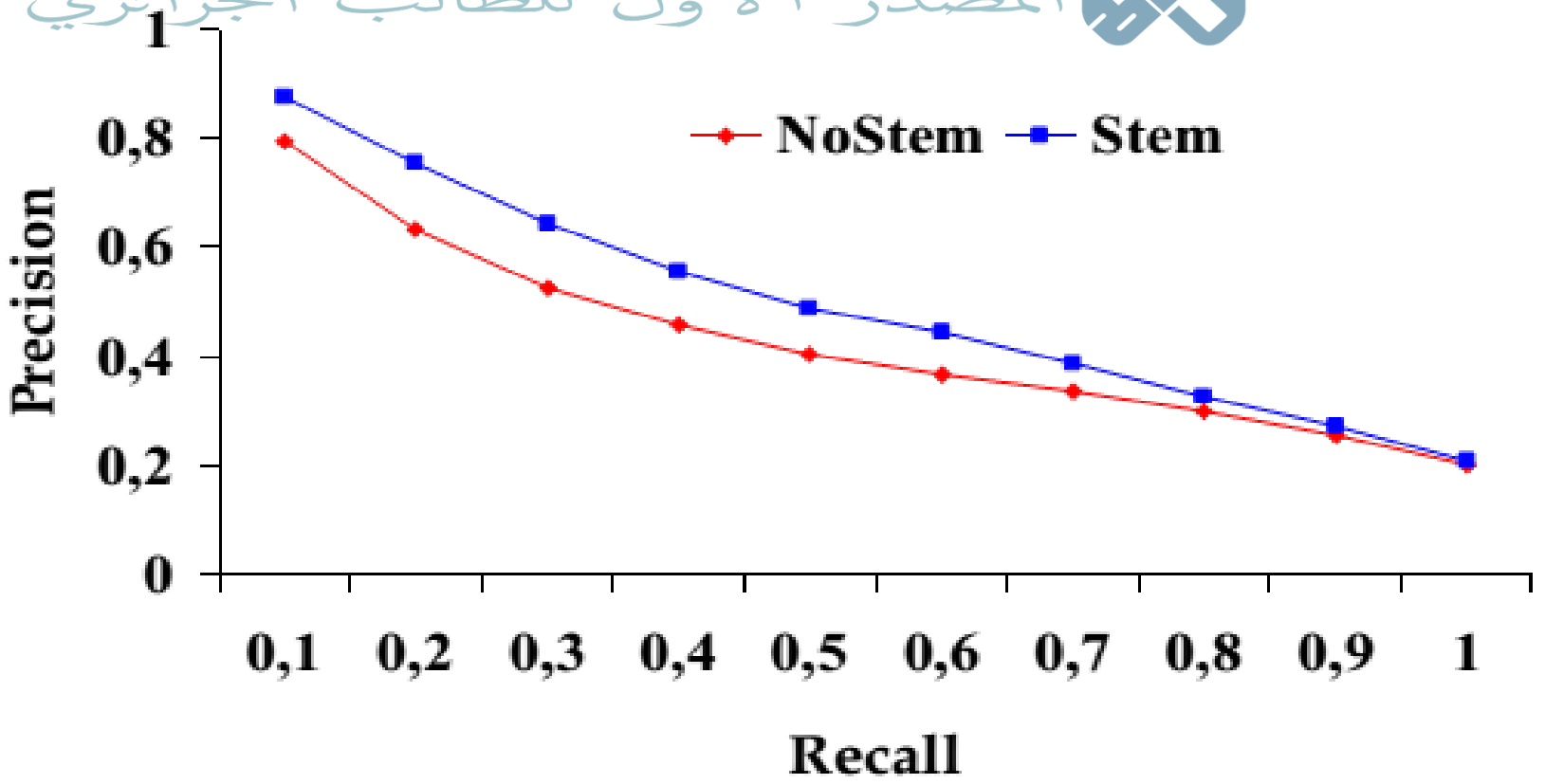
Ens des requêtes	
R	Pr
0	0,56535
0,1	0,3915
0,2	0,3205
0,3	0,24765
0,4	0,213
0,5	0,1837
0,6	0,1265
0,7	0,095
0,8	0,061
0,9	0,0335
1	0,0255
AvrPrec	0,1886

Exemple



Evaluation de performance dans les SRI

Comparaison de deux systèmes sur un ensemble de requêtes



Test N°2

Soient deux documents D1 et D2, représentés par les termes d'indexation T={t1, t2, t8}

Les poids des termes dans D1 et D2 sont:

t	t1	t2	t3	t4	t5	t6	t7	t8
W (ti ; D1)	0.5	0	0.7	1	0	0.3	0.6	0.8
W (ti ; D2)	0.2	0.3	0.5	0	0.8	0.4	0.9	0

Questions :

1- Donner les représentations de D1 et D2 dans le cas d'utilisation des modèles:

1) Booléen 2) Vectoriel

2- Soit la requête Q contenant les termes : **t1** et **t3** et **t6**

- Représenter et traiter cette requête selon les modèles booléen et vectoriel