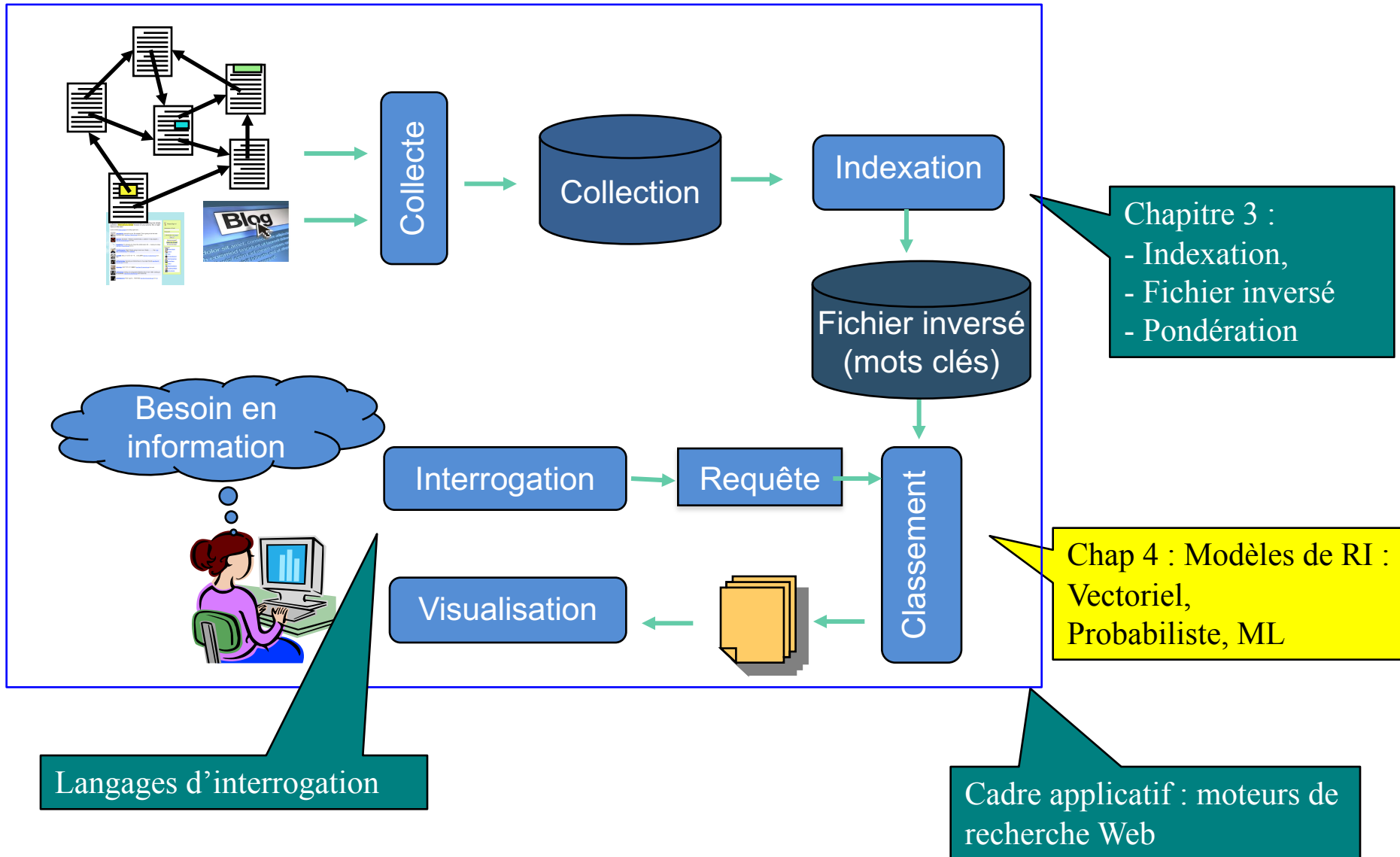


Chapitre 3 : Modèles de RI : booléen, vectoriel



Qu'est ce qu'un modèle de RI ?

- But : formalisation de la fonction de pertinence
- Les modèles de RI se distinguent par le principe d'appariement (*matching*) : appariement exact /approché (*Exact /Best matching*)
 - Appariement exact
 - Requête spécifie de manière précise les critères recherchés
 - L' ensemble des documents respectant exactement la requête sont sélectionnés, mais pas ordonné
 - Best matching (Ranking based models)
 - 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 ordonnés

IR models

- Appariement exact :
 - Théorie des ensembles :
 - Boolean model (± 1950)
- Modèle de tri de documents : Ranked models
 - Algèbre
 - Vector space model (± 1970)
 - Spreading activation model (± 1989)
 - LSI (Latent semantic Indexing)(± 1994)
 - Probabilité
 - Probabilistic model (± 1976)
 - Inference network model (± 1992)
 - Language model (± 1998)
 - *DFR (Divergence from Randomness model)* (± 2002)
 - Learning to rank

Appariement exact/Exact matching : Modèle booléen/Boolean Model

Le Modèle Booléen

- Le premier modèle de RI
- Basé sur la théorie des ensembles
- Un document est représenté un ensemble de termes
 - Ex : $d1(t1,t2,t5)$; $d2(t1,t3,t5,t6)$; $d3(t1,t2,t3,t4,t5)$
- Une requête est un ensemble de mots avec des opérateurs booléens : AND (\wedge), OR (\vee), NOT (\neg)
 - Ex: $q = t1 \wedge (t2 \vee \neg t3)$
- Appariement Exact basé sur la présence ou l'absence des termes de la requête dans les documents
 - Appariement $(q,d) = RSV(q,d)=1$ ou 0

Le Modèle Booléen

- $q = t1 \wedge (t2 \vee \neg t3)$
- $d1(t1,t2,t5); d2(t1,t3,t5,t6); d3(t1,t2,t3,t4,t5)$

Rsv(q,d1)=

Rsv(q,d2)=

Rsv(q,d3)=

Inconvénient du Modèle Booléen

- La sélection d'un document est basée sur une décision binaire
- Pas d'ordre pour les documents sélectionnés
- Formulation de la requête difficile pas toujours évidente pour beaucoup d'utilisateurs
- Problème de collections volumineuses : le nombre de documents retournés peut être considérable

Modèles de tri/ Rank-based models

Modèles de tri

- Plutôt que de renvoyer un ensemble de documents satisfaisant une requête booléenne, les modèles de tri retournent les documents dans un ordre trié censé représenter la pertinence de la requête vis-à-vis du document.
- Requêtes en texte libre: l'utilisateur exprime son besoin en fournissant au moteur de recherche une liste de mots clés
- Dans ces modèles on calcule un score de pertinence: RSV (requête, document).

Modèles de tri: score de pertinence

- Comment classer/trier les documents de la collection susceptibles de répondre à la requête?
- Attribuer un score - disons dans $[0, 1]$ - à chaque document
- Ce score mesure dans quelle mesure le document et la requête «correspondent» (« match »).

Modèles de tri: score de pertinence

- Assigner un poids à chaque terme du document (le poids est censé représenter l'importance du terme dans le document)
- Assigner (éventuellement) un poids à chaque terme de la requête (censé représenté l'intérêt que porte l'utilisateur au terme)
- $Score(q, d) = \sum_{(t \in q)} w(t, q) \cdot w(t, d)$

Poids du terme t ds
la requête

Poids du terme t ds
le document

Modèle de tri : pondération des termes

- *Le modèle tf.idf*

- *tf* : Idée sous jacente : plus un terme est fréquent dans un document plus il est important

$$tf = \begin{cases} freq(t,d) \\ 1 + \log(freq(t,d)) \\ \frac{freq(t,d)}{\max_{t' \in d} freq(t',d)} \\ \frac{freq(t,d)}{\sum_{t' \in d} freq(t',d)} \end{cases}$$

tion de ce

- Exemple de *tf*:

- “Okapi tf” : K introduit pour tenir

$$\frac{tf}{(K+tf)}$$

compte de la longueur des documents

$$tf = \frac{freq(t,d)}{k1.(1 - b + b * \frac{dl}{avgdl}) + freq(t,d)}$$

Taille (longueur)
du document

Modèle de tri : pondération tf.idf

- IDF : (Inverse Document Frequency) la fréquence du terme dans la collection

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

avec

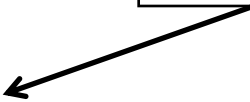
N : le nombre de documents de la collection,

n_t : le nombre de documents contenant le terme t

– Quelques formules répandues en RI

$$w(t,d) = tf * idf = \left\{ \begin{array}{l} \frac{(1 + \log(freq(t,d))) * \log \frac{N}{n_t}}{\sum_{\forall t' \in d} (1 + \log(freq(t',d))) * \log \frac{N}{n_{t'}}} \\ \frac{freq(t,d)}{k1.(1 - b + b * \frac{dl}{avgdl}) + freq(t,d)} * \log \frac{N - n_t}{n_t} \end{array} \right.$$

Facteur de normalisation



Modèle de tri

Modèle Vectoriel/Vector Space Model (VSM)

Modèle Vectoriel (*Vector Space Model*) (VSM)

- 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 :

$T \langle t_1, t_2, \dots, t_M \rangle$ (un terme = une dimension)

- Document : $d_j = (w_{1j}, w_{2j}, \dots, w_{Mj})$
- Requête : $q = (w_{1q}, w_{2q}, \dots, w_{Mq})$

w_{ij} : poids du terme t_i dans le document $d_j \rightarrow \text{tf} \cdot \text{idf}$

Modèle sac de mots

- La représentation vectorielle ne tient pas compte de l'ordre des mots
 - « Un garçon manque une pomme » est représenté par le même vecteur que « une pomme mange un garçon »
 - → c'est ce que l'on appelle « Sac de mots » (Bag of words)

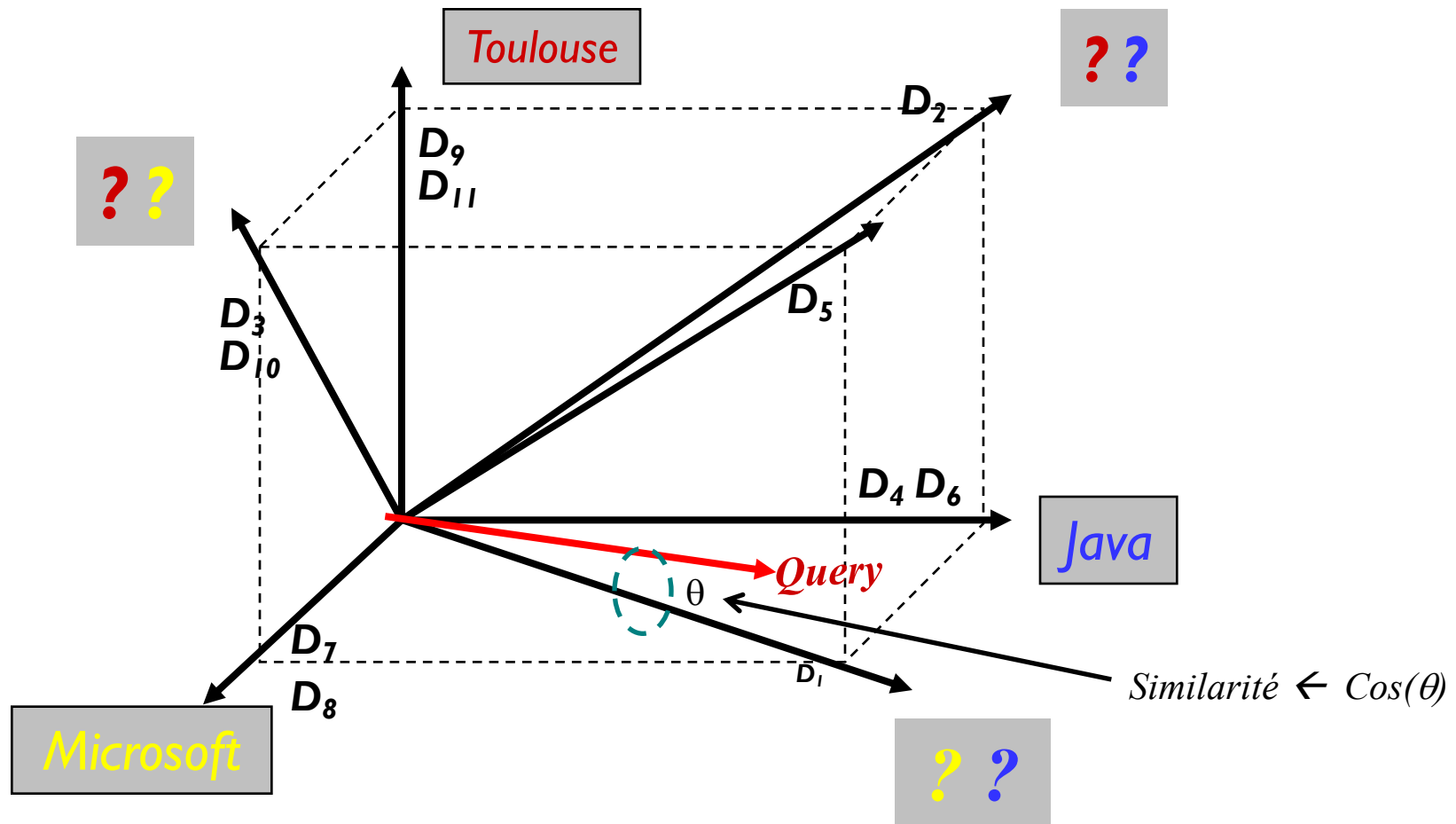
Modèle Vectoriel

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

$$\begin{pmatrix} & T_1 & T_2 & \dots & T_M \\ D_1 & w_{11} & w_{21} & \dots & w_{M1} \\ D_2 & w_{12} & w_{22} & \dots & w_{M2} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ D_n & w_{1n} & w_{2n} & \dots & w_{Mn} \end{pmatrix}$$

- La requête est également représentée par un vecteur.

illustration



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.

Similarité requête, document $\rightarrow \text{Cosine}(q,d)$

Dot product

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \bullet \vec{d}}{|\vec{q}| |\vec{d}|} = \frac{\vec{q}}{|\vec{q}|} \bullet \frac{\vec{d}}{|\vec{d}|} = \frac{\sum_{i=1}^{|V|} q_i d_i}{\sqrt{\sum_{i=1}^{|V|} q_i^2} \sqrt{\sum_{i=1}^{|V|} d_i^2}}$$

q_i est le poids du terme t_i dans la requête

d_i est le poids du terme t_i dans le document

Le Modèle Vectoriel

mesure de similarité

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

$$\frac{\|X \cap Y\|}{\|X\| + \|Y\| - \|X \cap Y\|}$$

$$\frac{\sum x_i * y_i}{\sum x_i^2 + \sum y_j^2 - \sum x_i * y_i}$$

Retour sur la pondération

→ tf*idf a plusieurs variantes

Term frequency		Document frequency		Normalization	
n (natural)	$tf_{t,d}$	n (no)	1	n (none)	1
l (logarithm)	$1 + \log(tf_{t,d})$	t (idf)	$\log \frac{N}{df_t}$	c (cosine)	$\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented)	$0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf)	$\max\{0, \log \frac{N-df_t}{df_t}\}$	u (pivoted unique)	$1/u$
b (boolean)	$\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$			b (byte size)	$1/CharLength^\alpha$, $\alpha < 1$
L (log ave)	$\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$				

Une variante est identifiée par un nom d'attribut pour chaque colonne (un tf, un idf, une normalisation)

Une pondération de type lnc → logarithme pour tf, pas d'idf, normalisation cosine
 Une pondération de type ltc → logarithme pour tf, idf et cosine

Dans le modèle vectoriel on aura ce type de notation :
ddd.qqq (ddd pour le document, qqq pour la requête)

$$score(q, d) = \sum_{t \in q} w(t, q) \cdot w(t, d)$$

Exemple Inc.ltc

Document: car insurance auto insurance

Query: best car insurance

Terme	Req (ltc)						Document(Inc)				Prod
	freq	tf	nd	idf	w(t,q)	Nor.li satio n	freq	tf-	w(t,d)	n' lisa tion	
auto	0	0	5000				1	1			
best	1	1	50000				0	0			
car	1	1	10000				1	1			
insurance	1	1	1000				2	1.3			

$N=10^6$ documents

Score (q,d)= 0.8

Suite exemple

- ddd.qqq=inc.ltc.

$$score(q, d) = \sum_{t \in q} \frac{(1 + \log(t, q)) * idf(t) * (1 + \log(t, d))}{\sqrt{\sum_{t \in q} ((1 + \log(t, q)) * idf(t))^2} \cdot \sqrt{\sum_{t \in d} (1 + \log(t, d))^2}}$$

Le Modèle Vectoriel

- Avantages:
 - La pondération améliore les résultats de recherche
 - La mesure de similarité permet d'ordonner les documents selon leur pertinence vis à vis de la requête
- Inconvénients:
 - La représentation vectorielle suppose l'indépendance entre termes (?)

Modèles de tri : Extension du modèle Booléen

Introduction

- 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 (extended Boolean Model)

Modèle booléen étendu

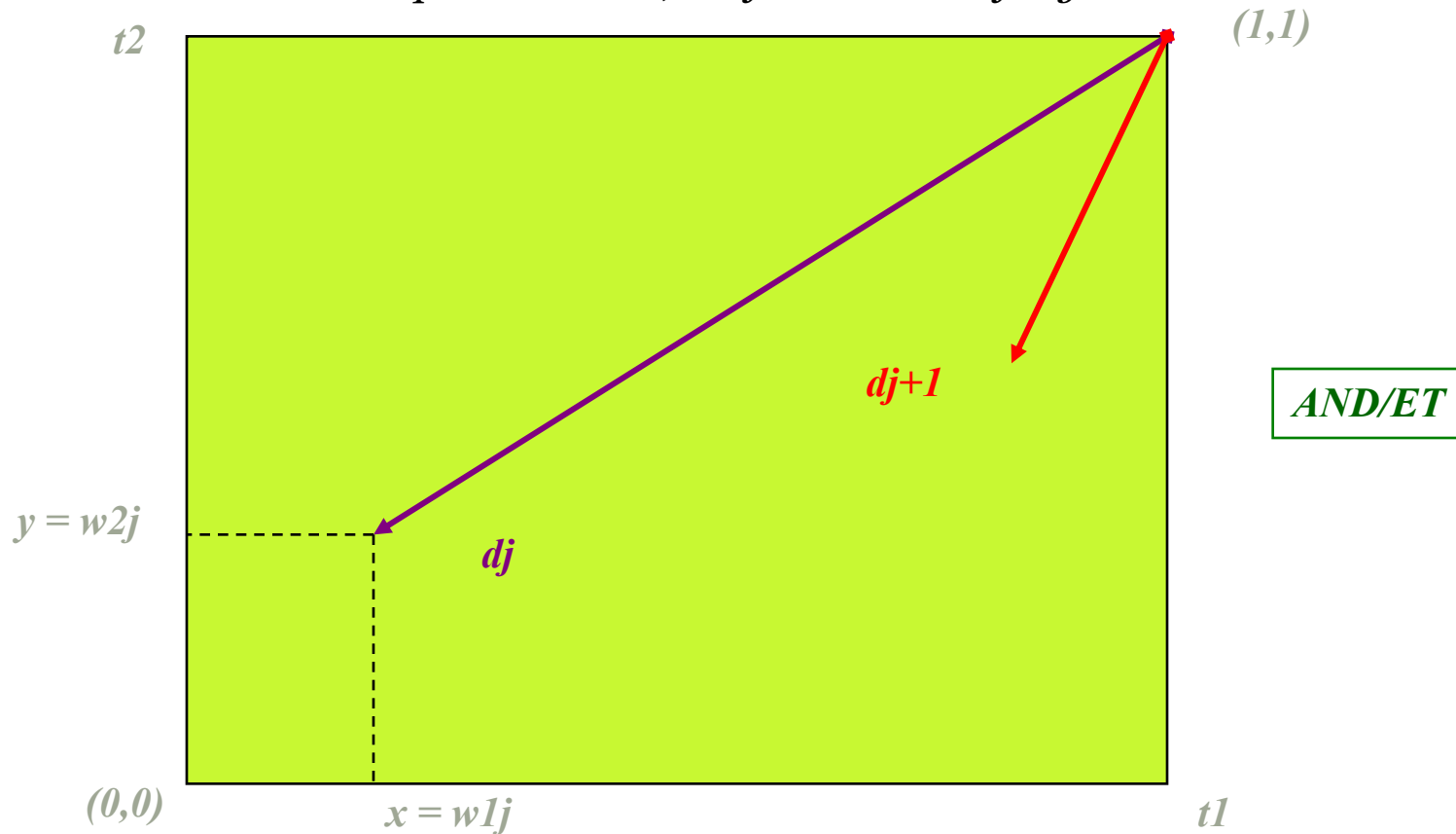
- Combinaison des modèles booléen et vectoriel
 - Document : liste de termes pondérés
 - Requête booléenne
 - Utilisation des distances algébriques pour mesurer la pertinence d' un document vis-à-vis à d' une requête

Modèle booléen étendu appariement

- Considérons
 - $d_j(w_{1j}, w_{2j}, \dots, w_{tj})$
 - q : requête à deux termes
 - $q_{\text{and}} = t_1 \text{ et } t_2$
 - $q_{\text{or}} = t_1 \text{ ou } t_3$

Intuition

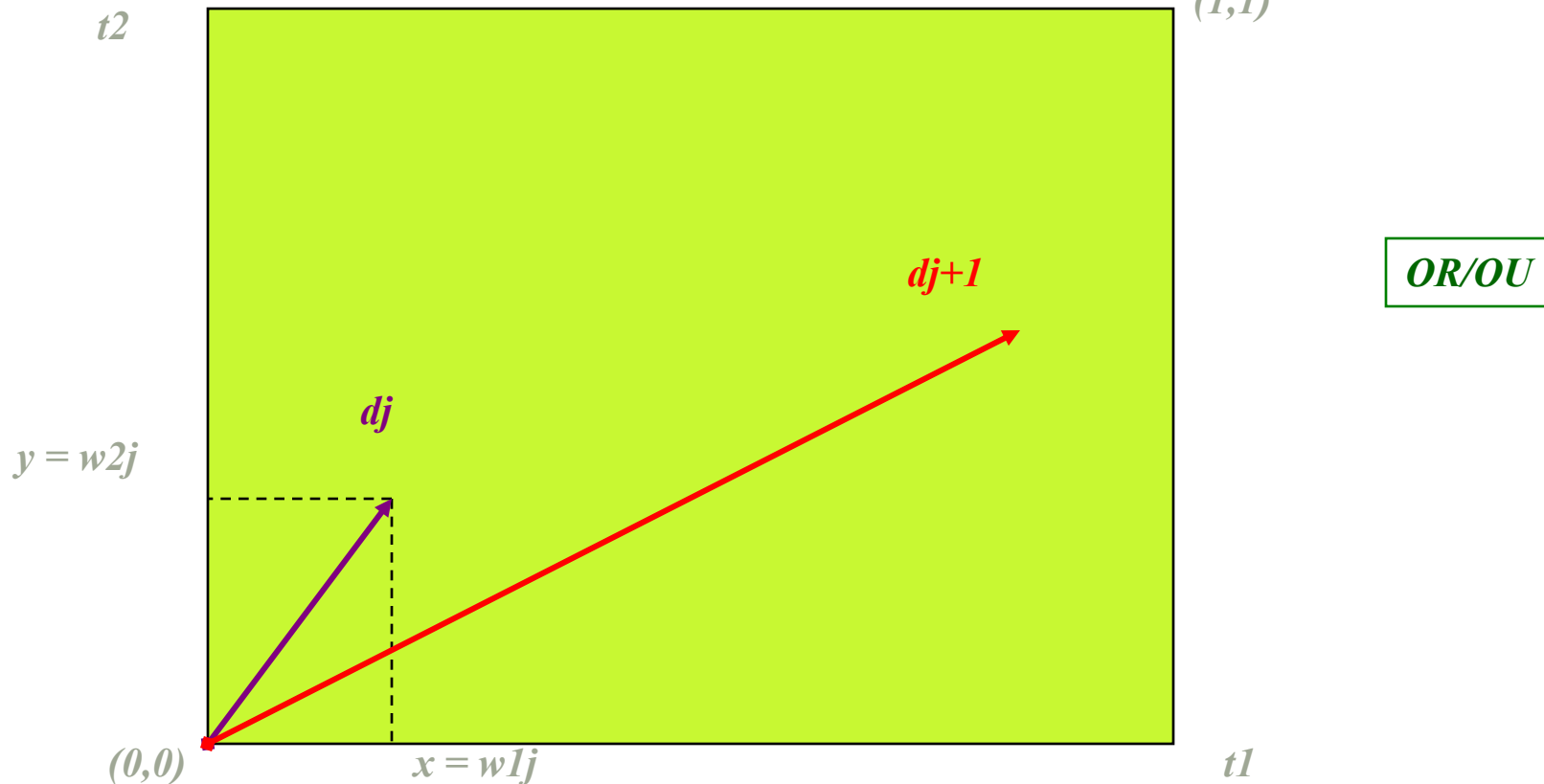
- $q \text{ and } = t_1 \wedge t_2; w_{1j} = x \text{ and } w_{2j} = y$



On veut se rapprocher du point $(1,1)$ $RSV(d_j, t_1 \wedge t_2) = 1 - \frac{\sqrt{((1-w_{1j})^2 + (1-w_{2j})^2)}}{\sqrt{2}}$

Intuition

- $qor = t_1 \vee t_2$; $wt_1 = x$ and $wt_2 = y$



On veut être le plus loin de (0,0)

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

Modèle booléen étendu appariement

- Considérons
 - $d_j (w_{1j}, w_{2j}, \dots, w_{tj})$
 - q : requête à deux termes

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

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

Modèle booléen (*pnorm*)étendu appariement

- Généralisation
 - Distance euclidienne à plusieurs dimensions
 - Utilisation de la **p-norm**
- Considérons :
 - un document d_j ($w_{1j}, w_{2j}, \dots, w_{tj}$) et
 - q (t_1, t_2, \dots, t_m) : une requête composée de **m** termes

$$RSV(d_j, q_{or}) = \left(\frac{w_{1j}^p + w_{2j}^p + \dots + w_{mj}^p}{m} \right)^{1/p}$$

$$RSV(d_j, q_{and}) = 1 - \frac{((1 - w_{1j})^p + (1 - w_{2j})^p + \dots + (1 - w_{mj})^p)^{1/p}}{m^{1/p}}$$

$$RSV(d_j, q_{not}) = 1 - RSV(d_j, q)$$

Modèle booléen(*p*norm) étendu appariement

- Si $p = 1$ alors (on retrouve le modèle vectoriel)
 - $RSV(d_j, q_{or}) = RSV(d_j, q_{and})$
- Si $p = \infty$ alors (modèle booléen)
 - $RSV(d_j, q_{or}) = \max (wx_j)$
 - $RSV(d_j, q_{and}) = \min (wx_j)$
- $p=2$ correspond à la distance euclidienne, semble être le meilleur choix

Modèle booléen (*pnorm*) étendu appariement

- Généralisation :
 - Si la requête et les documents sont pondérés
 - $q(q_1, q_2, \dots, q_m)$
 - $d_j (w_{1j}, w_{2j}, \dots, w_{tj})$

$$RSV(dj, qor) = \left(\frac{\sum q_i^p * w_{ij}^p}{\sum q_i^p} \right)^{1/p}$$

$$RSV(dj, qand) = 1 - \left(\frac{\sum q_i^p * (1 - w_{ij})^p}{\sum q_i^p} \right)^{1/p}$$

Modèle booléen étendu

- Modèle puissant
- Calcul complexe
- Problème de distributivité
 - $q_1 = (t_1 \text{ OU } t_2) \text{ ET } t_3$
 - $q_2 = (t_1 \text{ ET } t_3) \text{ OU } (t_2 \text{ ET } t_3)$
 - $\text{RSV}(q_1, d) \neq \text{RSV}(q_2, d)$

Exercice

- Exemple :
 - $T(\text{document, web, information, recherche, image, contenu})$: ensemble des termes d'indexation
 - $d1(\text{document } 0.3, \text{web } 0.5, \text{image } 0.2)$
 - $q1(\text{document OU web}); q2(\text{web ET document})$
 $q3((\text{web OU document}) \text{ ET image})$

Fin

- Pour ceux qui veulent aller plus loin(?), la suite porte sur des modèles basés sur la logique floue, très peu, voire, pas utilisés dans le domaine de la RI.

Ensembles flous (1.)

- Théorie des ensembles flous
 - Un cadre pour représenter les ensembles dont les bornes ne sont pas bien définis
 - L'objectif principal est l'introduction de la notion de degré d'appartenance d'un élément à un ensemble
 - Contrairement à la théorie des ensembles où un élément est dans l'ensemble ou ne l'est pas,
 - ...dans les ensembles flous, l'appartenance est mesurée par un degré variant entre 0 et 1
 - 0 → non appartenance
 - 1 → appartenance complète

Ensembles flous (2.)

- Définition

- Un sous ensemble A d'un univers de discours U est caractérisé par une fonction d'appartenance
 - $\mu_A: U \rightarrow [0,1]$
 - qui associe à chaque élément u de U un nombre $\mu_A(u)$ dans $[0,1]$
- Soient A et B deux sous-ensembles flous de U

- Complément $\mu_A(u)$ $\mu_{\bar{A}}(u) = 1 - \mu_A(u)$

- Union $\mu_{A \cup B}(u) = \max(\mu_A(u), \mu_B(u))$

- Intersection $\mu_{A \cap B}(u) = \min(\mu_A(u), \mu_B(u))$

Modèle flou de RI

- Un document est un ensemble de termes
- chaque terme à un poids qui mesure à quel point le terme caractérise le document
- Ces poids sont dans $[0, 1]$. (dans le booléen standard un terme est soit présent 1 ou absent 0 dans un document)
- On pourrait écrire :
$$\mu_d(t) = w_{dt}$$

Modèle flou de base, requête non pondérée

- Soient :
 - Termes: t_1, t_2, \dots, t_n
 - Document: $d(w_1, w_2, \dots, w_n)$
- *Requête disjonctive* : $q_{\text{or}} = (t_1 \vee t_2 \vee \dots \vee t_n)$
 - $\text{RSV}(q_{\text{or}}, d) := \max(w_1, w_2, \dots, w_n)$
- *Requête conjonctive* : $q_{\text{and}} = (t_1 \wedge t_2 \wedge \dots \wedge t_n)$
 - $\text{RSV}(q_{\text{and}}, d) = \min(w_1, \dots, w_n)$
- Généralisation
 - $\text{RSV}(d, q1 \wedge q2) = \min(\text{RSV}(d, q1), \text{RSV}(d, q2))$,
 - $\text{RSV}(d, q1 \vee q2) = \max(\text{RSV}(d, q1), \text{RSV}(d, q2))$,

Modèle flou requête pondérée

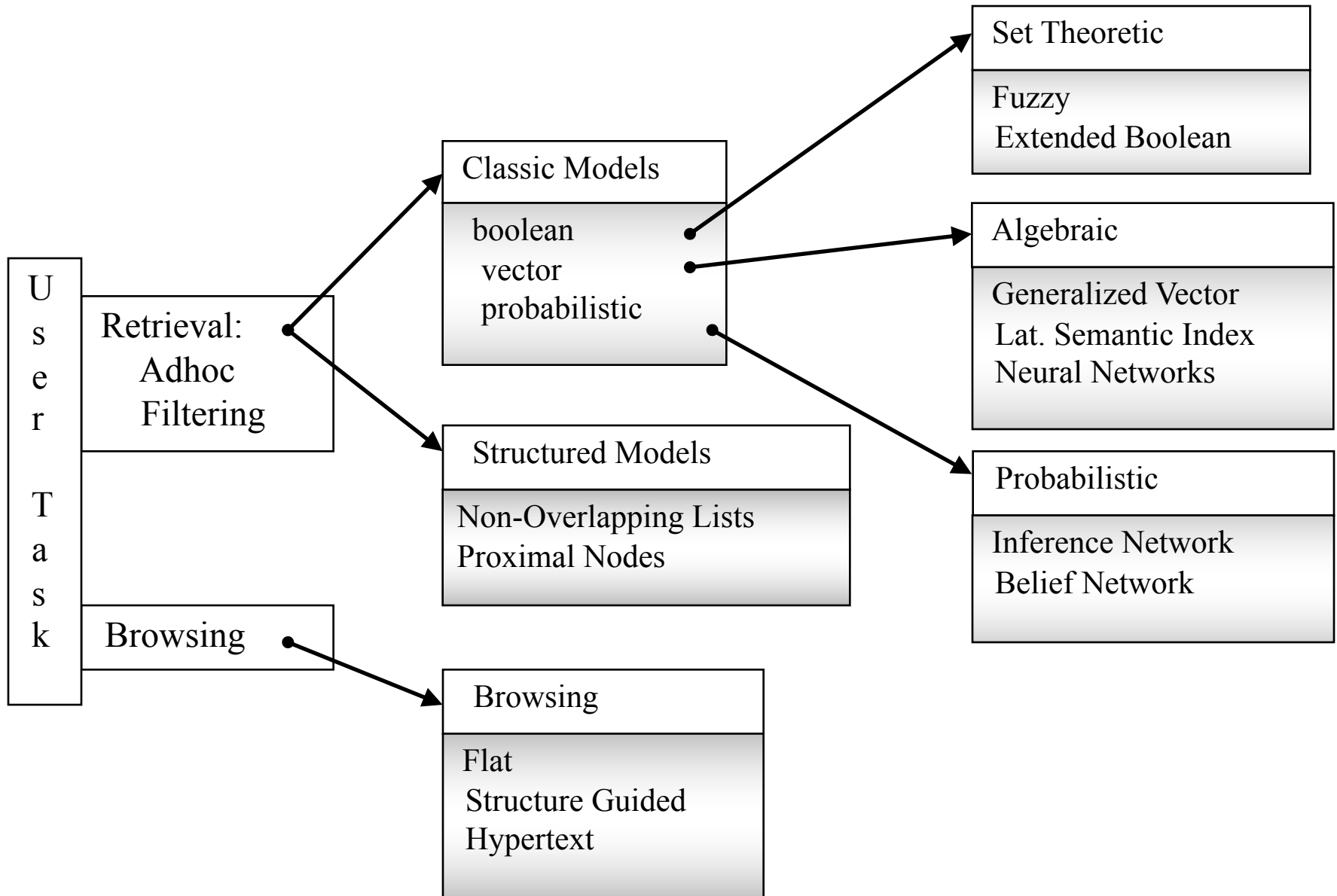
- Requête à avec un terme unique

$$g(F(d, t), a) = \begin{cases} \min(a, F(d, t)) & \text{si disjonction} \\ \max(1 - a, F(d, t)) & \text{si conjonction} \end{cases}$$

- Autres opérateurs
 - L'implication de Dienes : $a \rightarrow b = \max(1 - a, b)$.
 - L'implication de Gödel : $a \rightarrow b = 1$ si $a \leq b$ $a \rightarrow b = b$ si $a > b$.
 - L'implication de Lukasiewicz : $a \rightarrow b = \min(1, 1 - a + b)$.
- Requête à plusieurs termes :
 - 1-Agréger les termes 1 à 1 selon un des opérateurs ci dessus
 - 2- Agréger toute la requête min ou max

Compléments du cours

Modèles de RI



Modèles de RI

