



# RECHERCHE D'INFORMATION

## INFORMATION RETRIEVAL

CHAPITRE 4: Modèles booléen, Vectoriel

13 octobre 2025

Modèles de base de la RI

### I. INTRODUCTION

L'objectif d'un modèle de RI est de formaliser le processus de recherche d'information. Un modèle est une abstraction d'un processus.

Un modèle de RI doit comporter au minimum les modules suivants :

- ✓ Un module de représentation des documents (indexation)
- ✓ Un module de représentation des requêtes
- ✓ Un module d'appariement entre un document et une requête (similarité)

Modèles de base de la RI

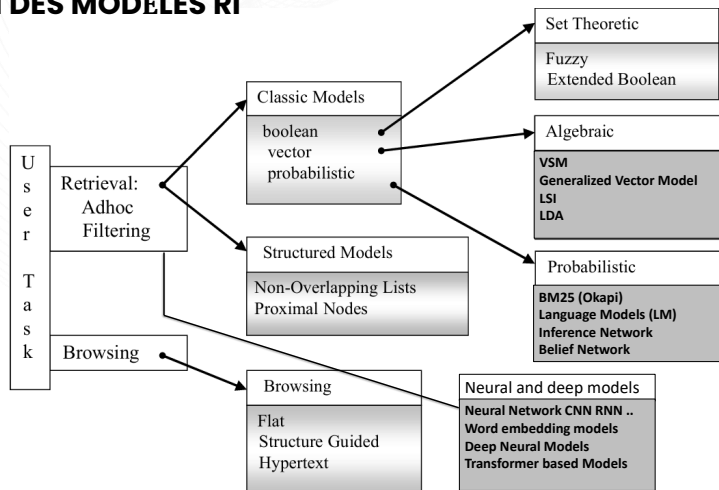
### II. CLASSIFICATION DES MODÈLES RI

**Set Theoretic Models** Represent documents and queries as **sets of terms**; retrieval based on set operations.

**Algebraic / Vector-Based Models** Represent documents and queries as **vectors or matrices**; use algebraic similarity functions.

**Probabilistic Models** Rank documents by the **probability of being relevant** to a given query.

**Neural and deep models** Use **neural networks** to learn document-query relationships directly from data



Modèles de base de la RI

### III. MODÈLES DE BASE À ÉTUDIER

Dans ce chapitre nous allons étudier les modèles de base suivants :

- ✓ **Modèle booléen de base**
- ✓ **Modèle vectoriel**
- ✓ **Modèle booléen basé sur les ensembles flous**
- ✓ **Modèle booléen étendu**
- ✓ **Modèle booléen P-norme étendu**

**Set Theoretic Model**

**Algebraic Model**

**Set Theoretic Model**

**Set Theoretic Model**

**Set Theoretic Model**

### III.1. MODÈLE BOOLÉEN – BOOLEAN MODEL

- ✓ Le premier modèle formel proposé dans le domaine de la RI
- ✓ Repose sur la théorie des ensembles et la logique booléenne pour décrire la relation entre document et requêtes

#### A. Module de représentation des documents

- Dans ce modèle, chaque document est représenté par un ensemble de termes.
- Un terme a un poids binaire: 1 s'il est présent dans le document, 0 sinon.
- Aucune pondération (comme la fréquence ou le poids TF-IDF) n'est calculée.

Le modèle ne tient donc pas compte de la fréquence d'apparition d'un terme, ni de sa rareté dans la collection.

### III.1. MODÈLE BOOLÉEN – BOOLEAN MODEL

#### A. Module de représentation des documents

- Chaque document est représenté comme un vecteur binaire dans un espace de termes. La valeur 1 ou 0 indique la présence ou l'absence de ce terme dans le document.

$$d_j = (w_{1j}, w_{2j}, \dots, w_{nj}) \quad w_{ij} = \begin{cases} 1 & \text{si } t_i \in d_j \\ 0 & \text{sinon} \end{cases}$$

Exemple : d1 contient les termes : t1,t5,t7 est représenté par d1(1,0,0,0,1,0,1)

### III.1. MODÈLE BOOLÉEN – BOOLEAN MODEL

#### B. Module de représentation de query

Une requête est un ensemble de mots exprimée sous forme logique combiné à l'aide des opérateurs booléens : **AND (∧), OR(∨), NOT (¬)**

- **AND (∧)** : intersection — le document contient **tous** les termes reliés par AND.
- **OR (∨)** : union — le document contient **au moins un** des termes reliés par OR.
- **NOT (¬)** : négation — le document **ne contient pas** le terme indiqué.

Interprétation :

- Le document doit inclure le terme  $t_1$ ,
- et soit inclure  $t_2$  soit ne pas contenir  $t_3$ .

Exemple: query =  $t_1 \wedge (t_2 \vee \neg t_3)$

### III.1. MODÈLE BOOLÉEN – BOOLEAN MODEL

#### C. Module d'appariement / Matching

La similarité entre un document et une requête est calculée par une valeur exacte basée sur la présence ou l'absence des termes de la requête dans les documents, qui est soit 1 soit 0.

On note l'appariement (q,d) par **RSV(q,d)** qui signifie « Retrieval Status Value » avec q: query et d: Document et Indique la pertinence du document  $d$  par rapport à la requête  $q$ .

- 1) **RSV(q,d) = 1** si en remplaçant les termes dans la requête par leurs poids dans le document (0 ou 1), puis en évaluant cette requête comme une expression logique, elle donnera 1
- 2) **RSV(q,d) = 0** sinon

### III.1. MODÈLE BOOLÉEN – BOOLEAN MODEL

#### EXEMPLE

Soit l'ensemble des termes d'indexation  
(document, web, information, recherche, image, contenu).

#### Documents :

- $d_1$  = (document web document web document)
- $d_2$  = (image contenu web)
- $d_3$  = (document recherche information)

#### Requêtes :

- $q_1$  = (document  $\wedge$  web)  $\vee$  image
- $q_2$  = (document  $\vee$  web)  $\wedge$  image
- $q_3$  = (web  $\vee$  image)  $\wedge$  document

#### Représentation binaire des documents

Terme	$d_1$	$d_2$	$d_3$
document	1	0	1
web	1	1	0
information	0	0	1
recherche	0	0	1
image	0	1	0
contenu	0	1	0

### III.1. MODÈLE BOOLÉEN – BOOLEAN MODEL

#### EXEMPLE : Évaluation des requêtes

Document	document	web	image	$q_1 = (\text{document} \wedge \text{web}) \vee \text{image}$	$q_2 = (\text{document} \vee \text{web}) \wedge \text{image}$	$q_3 = (\text{web} \vee \text{image}) \wedge \text{document}$
$d_1$	1	1	0	$(1 \wedge 1) \vee 0 = 1$	$(1 \vee 1) \wedge 0 = 0$	$(1 \vee 0) \wedge 1 = 1$
$d_2$	0	1	1	$(0 \wedge 1) \vee 1 = 1$	$(0 \vee 1) \wedge 1 = 1$	$(1 \vee 1) \wedge 0 = 0$
$d_3$	1	0	0	$(1 \wedge 0) \vee 0 = 0$	$(1 \vee 0) \wedge 0 = 0$	$(0 \vee 0) \wedge 1 = 0$

Pour  $q_1$  les  $d_1$  et  $d_2$  sont pertinent , pour  $q_2$  seulement le  $d_2$  est pertinent , pour  $q_3$  seul le  $d_1$  est pertinent

### III.1. INCONVÉNIENT DU MODÈLE BOOLÉEN

- ✓ La sélection d'un document se fait sur la base d'une décision **binaire** (pertinent / non pertinent).
- ✓ Aucun **ordre de pertinence** n'est établi entre les documents sélectionnés.
- ✓ La **formulation des requêtes** est souvent difficile et pas toujours intuitive pour les utilisateurs.
- ✓ Dans le cas de **collections volumineuses**, le nombre de documents retournés peut être très important.

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### Origine :

- Proposé par *Gérard Salton* dans le système SMART (1970).
- Basé sur les principes de l'algèbre vectorielle.

#### Idée clé :

- Représenter documents et requêtes comme des vecteurs dans un espace de termes.
- Mesurer leur proximité géométriquement pour estimer la pertinence.

#### Pourquoi :

Le modèle booléen est trop rigide : il répond par **oui/non**.

- Le modèle vectoriel introduit une notion de degré de similarité.
- Il permet de classer les documents selon leur pertinence par rapport à une requête.
- On passe d'une recherche exacte à une recherche graduelle (ranking).

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### A. Module de représentation des documents

- Chaque document est représenté sous forme de vecteur de poids dans l'espace vectoriel engendré par tous les termes de la collection de documents.

$$T = \langle t_1, t_2, \dots, t_M \rangle \quad \text{où :}$$

- $M$  = nombre total de termes dans la collection

$$d_j = (w_{1j}, w_{2j}, \dots, w_{Mj})$$

- $w_{ij}$  = poids du terme  $t_i$  dans le document  $d_j$

- Chaque terme est pondéré selon une formule de pondération. La plus fréquemment utilisé

$$w_{ij} = TF_{ij} \times IDF_i$$

- Donc une collection est représentée par un fichier inverse avec pondération (vu dans le chapitre précédent)

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### B. Module de représentation de query

- Les requêtes sont aussi représentées sous forme de vecteurs dans l'espace vectoriel engendré par tous les termes de la collection de documents.

- On note  $Wiq$  le poids du terme dans la requête  $q$

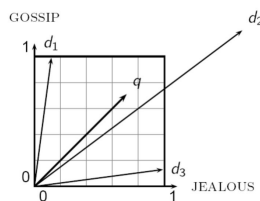
- Chaque terme est pondéré par  $Wiq = 1$  s'il existe dans la requête, 0 sinon.

- Donc, une requête  $q = (w_{1q}, w_{2q}, \dots, w_{Mq})$  avec  $M$  le nombre de termes dans la collection.

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### C. Module d'appariement / Matching

- La pertinence dans le modèle vectoriel d'un document par rapport à une requête est mesurée à travers une similarité vectorielle.
- Chaque document et chaque requête sont représentés sous forme de vecteurs dans un espace de termes.



- La similarité dépend de l'angle entre ces vecteurs.
- Plus l'orientation du vecteur document est proche de celle du vecteur requête, plus le document est jugé pertinent.
- La similarité est mesurée par le cosinus de l'angle  $\theta$ .

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### C. Module d'appariement / Matching

Pourquoi le cosinus entre les vecteurs (requête et document)

Rappel algébrique:

- Le cosinus mesure la **proportion de direction commune** entre deux vecteurs

$$\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \|\vec{v}\|} \quad \cos(\theta) = \frac{u_1 v_1 + u_2 v_2}{\sqrt{u_1^2 + u_2^2} \sqrt{v_1^2 + v_2^2}}$$

- **Le Produit scalaire :  $u_1 v_1 + u_2 v_2$**

mesure à quel point les deux vecteurs vont dans la même direction.

- **Les normes  $\|\vec{u}\|$  et  $\|\vec{v}\|$**  servent à neutraliser la taille des vecteurs (on compare juste leur orientation).

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### C. Module d'appariement / Matching

Donc la formule de similarité entre document et requête s'adopte comme suit dans le modèle vectoriel :

$$\text{Sim}(q, d_j) = \cos(\theta) = \frac{q \cdot d_j}{\|q\| \|d_j\|}$$

#### Pourquoi normaliser les vecteurs ?

- Les documents longs ont plus de termes et donc des poids plus grands.
- Pour éviter ce biais, on divise par la norme du vecteur.
- Ainsi, la similarité dépend uniquement de la direction, pas de la longueur.

Avec :

$$q \cdot d_j = \sum_i w_{iq} \times w_{ij}$$

$$\|q\| = \sqrt{\sum_i w_{iq}^2}, \quad \|d_j\| = \sqrt{\sum_i w_{ij}^2}$$

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### MESURES DE SIMILARITÉ VECTORIELLE

La similarité entre un document et une requête est calculée selon l'une des formules suivantes :

**Produit interne**  $RSV(d_j, q) = \sum w_{ij} * w_{iq}$

**Coef. de Dice**  $RSV(d_j, q) = \frac{2 * \sum w_{ij} * w_{iq}}{\sum w_{ij}^2 + \sum w_{iq}^2}$

**Mesure du cosinus**  $RSV(d_j, q) = \frac{\sum w_{ij} * w_{iq}}{\sqrt{\sum w_{ij}^2 * \sum w_{iq}^2}}$

**Mesure du Jaccard**  $RSV(d_j, q) = \frac{\sum w_{ij} * w_{iq}}{\sum w_{ij}^2 + \sum w_{iq}^2 - \sum w_{ij} * w_{iq}}$

$(w_{i,j})$  poids du terme  $i$  dans le document  $j$   
 $(w_{i,q})$  poids du terme  $i$  dans la requête  $q$ .

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

Soit la collection de 3 documents suivants :

D1 : { langage de programmation python est très utilisé pour le traitement de texte }

D2 : { le langage JAVA est basé sur le langage C++ }

D3 : { un langage de programmation est un langage utilisé pour traduire un algorithme en un programme }

stopwords: { de, est, très, pour, le, un, en }

La requête **q** : { langage python java }

1. Donner le fichier inverse de la collection avec la formule :  $\text{poids}(ti, dj) = (\text{freq}(ti, dj) / \max(\text{freq}(dj))) * \log(N/n_i + 1)$
2. Calculer la similarité entre chaque document et la requête  $q$  par les quatre formules du modèle vectoriel

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### Données (rappel)

Documents (après suppression des stopwords) :

• D1 = { langage, programmation, python, utilisé, traitement, texte }

• D2 = { langage, JAVA, basé, langage, C++ }

• D3 = { langage, programmation, langage, utilisé, traduire, algorithme, programme }

Requête :  $q = \{\text{langage, python, java}\}$  on prend  $\text{java} \equiv \text{JAVA}$

Nombre de documents  $N = 3$ .

Les poids (déjà calculés)

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE :

Terme	D1	D2	D3
langage	0.301	0.301	0.301
programmation	0.398	0	0.199
python	0.602	0	0
utilisé	0.398	0	0.199
traitement	0.602	0	0
texte	0.602	0	0
JAVA	0	0.301	0
basé	0	0.301	0
C++	0	0.301	0
traduire	0	0	0.301
algorithme	0	0	0.301
programme	0	0	0.301

Produit interne

$$RSV(d_j, q) = \sum w_{ij} * w_{iq}$$

Coef. de Dice

$$RSV(d_j, q) = \frac{2 * \sum w_{ij} * w_{iq}}{\sum w_{ij}^2 + \sum w_{iq}^2}$$

Mesure du cosinus

$$RSV(d_j, q) = \frac{\sum w_{ij} * w_{iq}}{\sqrt{\sum w_{ij}^2 * \sum w_{iq}^2}}$$

Mesure du Jaccard

$$RSV(d_j, q) = \frac{\sum w_{ij} * w_{iq}}{\sum w_{ij}^2 + \sum w_{iq}^2 - \sum w_{ij} * w_{iq}}$$

 $(w_{i,j})$  poids du terme  $i$  dans le document  $j$  $(w_{i,q})$  poids du terme  $i$  dans la requête  $q$ .Indice: Requête  $q = \{\text{langage, python, java}\}$ 

Réaliser une représentation vecteur query binaire

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### 1) Vocabulaire (ordre des dimensions)

On reprend le vocabulaire utilisé pour la collection (même ordre que précédemment)

#### 2) Requête $q = \{\text{langage, python, java}\}$ — représentation vecteur binaire

Règle :  $w_{i,q} = 1$  si le terme  $t_i$  appartient à la requête, sinon 0.

Donc le **vecteur requête** (même ordre que le vocabulaire) est :

$$q = (1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0)$$

On retient généralement seulement les composantes non nulles :  $q$  contient les dimensions

« langage, python, JAVA ».

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### 3) Calcul de Similarité entre la query q et chaque document

##### • Produit interne

$$\bullet RSV(D1, q) = 0.301_{(\text{langage})} + 0.602_{(\text{python})} + 0_{(\text{JAVA})} = 0.903$$

$$\bullet RSV(D2, q) = 0.301_{(\text{langage})} + 0_{(\text{JAVA})} + 0_{(\text{python})} = 0.602$$

$$\bullet RSV(D3, q) = 0.301_{(\text{langage})} + 0 + 0 = 0.301$$

Classement : **D1 > D2 > D3**

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### 3) Calcul de Similarité entre la query q et chaque document

##### • Mesure de Cosinus

$$\text{Norme de la requête } \sqrt{\sum_i w_{i,q}^2} = \sqrt{1^2 + 1^2 + 1^2} = \sqrt{3} = 1.732$$

$$\begin{aligned} \text{Document D1 : } \sqrt{\sum_{i=1}^8 w_{i,1}^2} &= \sqrt{0.301^2 + 0.398^2 + 0.602^2 + 0.398^2 + 0.602^2 + 0.602^2} \\ &= \sqrt{1.4946} = 1.223 \\ RSV(D1, q) &= \frac{0.903}{1.223 \times 1.732} = 0.426 \end{aligned}$$



### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

3) Calcul de Similarité entre la query q et chaque document

- Mesure de Cosinus

Norme de la requête  $\sqrt{\sum_i w_{i,q}^2} = \sqrt{1^2 + 1^2 + 1^2} = \sqrt{3} = 1.732$

Document D2 :  $\sqrt{\sum_{i=1}^8 w_{i,2}^2} = \sqrt{0.301^2 + 0.301^2 + 0.301^2 + 0.301^2}$   
 $= \sqrt{0.3624} = 0.602$   
 $RSV(D2, q) = \frac{0.602}{0.602 \times 1.732} = 0.577$

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

3) Calcul de Similarité entre la query q et chaque document

- Mesure de Cosinus

Norme de la requête  $\sqrt{\sum_i w_{i,q}^2} = \sqrt{1^2 + 1^2 + 1^2} = \sqrt{3} = 1.732$

Document D3 :  $\sqrt{\sum_{i=1}^8 w_{i,3}^2} = \sqrt{0.301^2 + 0.199^2 + 0.199^2 + 0.301^2 + 0.301^2 + 0.301^2}$   
 $= \sqrt{0.4416} = 0.665$   
 $RSV(D, 3q) = \frac{0.301}{0.665 \times 1.732} = 0.262$

Classement : D2 > D1 > D3

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

3) Calcul de Similarité entre la query q et chaque document

- Coef de Dice  $\sum_i w_{i,q}^2 = 1^2 + 1^2 + 1^2 = 3.$

Document D1 :

Somme des carrés (document) :

$$\sum_i w_{i,1}^2 = 0.301^2 + 0.398^2 + 0.602^2 + 0.398^2 + 0.602^2 + 0.602^2 = 1.494621$$

$$\sum_i w_{i,1} \cdot w_{i,q} = 0.903.$$

Donc :

$$\text{Dice}(D1, q) = \frac{2 \times 0.903}{1.494621 + 3} = \frac{1.806}{4.494621} \approx 0.4018$$

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

3) Calcul de Similarité entre la query q et chaque document

- Coef de Dice  $\sum_i w_{i,q}^2 = 1^2 + 1^2 + 1^2 = 3.$

Document D2 :

Somme des carrés (document) :

$$\sum_i w_{i,2}^2 = 0.301^2 + 0.301^2 + 0.301^2 + 0.301^2 = 0.362404$$

$$\sum_i w_{i,2} \cdot w_{i,q} = 0.602.$$

Donc :

$$\text{Dice}(D2, q) = \frac{2 \times 0.602}{0.362404 + 3} = \frac{1.204}{3.362404} \approx 0.3581$$

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### 3) Calcul de Similarité entre la query q et chaque document

- Coef de Dice  $\sum_i w_{i,q}^2 = 1^2 + 1^2 + 1^2 = 3$ .

Document D3 :

$$\sum_i w_{i,3}^2 = 0.301^2 + 0.199^2 + 0.199^2 + 0.301^2 + 0.301^2 + 0.301^2 = 0.441606$$

$$\sum_i w_{i,3} \cdot w_{i,q} = 0.301.$$

Donc :

$$\text{Dice}(D, 3q) = \frac{2 \times 0.301}{0.441606 + 3} = \frac{0.602}{3.441606} \approx 0.1749$$

**Classement D1 > D2 > D3**

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### 3) Calcul de Similarité entre la query q et chaque document

- Mesure de Jaccard

$$\text{Document D1 : } \sum_i w_{i,1}^2 = 0.301^2 + 0.398^2 + 0.602^2 + 0.398^2 + 0.602^2 + 0.602^2 = 1.494621$$

$$\sum_i w_{i,1} \cdot w_{i,q} = 0.903$$

$$\text{Jaccard}(D, 1q) = \frac{0.903}{1.494621 + 3 - 0.903} = \frac{0.903}{3.591621} \approx 0.25142$$

$$\text{Document D2 : } \sum_i w_{i,2}^2 = 0.301^2 + 0.301^2 + 0.301^2 + 0.301^2 = 0.362404$$

$$\sum_i w_{i,2} \cdot w_{i,q} = 0.602$$

$$\text{Jaccard}(D, 2q) = \frac{0.602}{0.362404 + 3 - 0.602} = \frac{0.602}{2.760404} \approx 0.21808$$

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### EXEMPLE PRATIQUE

#### 3) Calcul de Similarité entre la query q et chaque document

- Mesure de Jaccard

$$\text{Document D3 : } \sum_i w_{i,3}^2 = 0.301^2 + 0.199^2 + 0.199^2 + 0.301^2 + 0.301^2 + 0.301^2 = 0.441606$$

$$\sum_i w_{i,3} \cdot w_{i,q} = 0.301$$

$$\text{Jaccard}(D3, q) = \frac{0.301}{0.441606 + 3 - 0.301} = \frac{0.301}{3.140606} \approx 0.09584$$

**Classement D1 > D2 > D3**

### III.2. MODÈLE VECTORIEL – VECTOR SPACE MODEL (VSM)

#### AVANTAGES DU MODÈLE VECTORIEL

- ✓ 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
- ✓ Simple à programmer

#### INCONVÉNIENTS DU MODÈLE VECTORIEL

- La représentation vectorielle suppose l'indépendance entre termes
- Le sens des termes n'est pas pris en compte



### III.3. MODÈLE BOOLÉEN BASÉ SUR LES ENSEMBLES FLOUS

#### Origine et Principe

- Ce modèle est une extension du modèle booléen classique.
- Introduit la notion de degré d'appartenance au lieu d'une valeur binaire (0 ou 1).
- Un document peut être partiellement pertinent pour une requête.
- Ce modèle relie la logique booléenne à la théorie des ensembles flous (fuzzy sets) de Zadeh, 1965.

### III.3. MODÈLE BOOLÉEN BASÉ SUR LES ENSEMBLES FLOUS

#### A. Module de représentation des documents

- Chaque document est représenté comme un ensemble de termes avec des poids flous, qui mesure à quel point le terme caractérise le document.

$$d_j = \{(t_1, w_{1j}), (t_2, w_{2j}), \dots, (t_M, w_{Mj})\}$$

où :

- $t_i$  : terme de la collection
- $w_{ij} \in [0, 1]$  : degré d'appartenance du terme  $t_i$  au document  $d_j$

- Ces poids proviennent souvent du **TF-IDF** ou d'autres mesures de pondération

### III.3. MODÈLE BOOLÉEN BASÉ SUR LES ENSEMBLES FLOUS

#### B. Représentation des requêtes

Une requête reste booléenne mais appliquée à des valeurs floues. Ex:  $q = t_1 \wedge (t_2 \vee \neg t_3)$

#### C. Calcul de la similarité floue

Le degré de pertinence RSV entre la requête et un document est calculée selon les formules suivantes :

$$RSV(d_j, t_i) = w_{ij}$$

$$RSV(d_j, t_1 \wedge t_2) = \min(RSV(d_j, t_1), RSV(d_j, t_2))$$

$$RSV(d_j, t_1 \vee t_2) = \max(RSV(d_j, t_1), RSV(d_j, t_2))$$

$$RSV(d_j, \neg t_i) = 1 - RSV(d_j, t_i)$$

### III.3. MODÈLE BOOLÉEN BASÉ SUR LES ENSEMBLES FLOUS

#### Exemple pratique

Supposons un document  $d_1$  avec les poids suivants (issus d'un calcul TF\*IDF ou donnés) :

- $RSV(d_1, t_1) = 0.8$
- $RSV(d_1, t_2) = 0.4$
- $RSV(d_1, t_3) = 0.6$

**Conjonction (ET)**  $RSV(d_1, t_1 \wedge t_2) = \min(0.8, 0.4) = 0.4$

**Disjonction (OU)**  $RSV(d_1, t_1 \vee t_2) = \max(0.8, 0.4) = 0.8$

**Négation (NON)**  $RSV(d_1, \neg t_3) = 1 - 0.6 = 0.4$

- Le document est **pertinent à 0.4** pour la requête  $t_1 \wedge t_2$ .
- Il est **pertinent à 0.8** pour la requête  $t_1 \vee t_2$ .
- Il est **peu pertinent (0.4)** pour la requête  $\neg t_3$ .

### III.3. MODÈLE BOOLÉEN BASÉ SUR LES ENSEMBLES FLOUS

#### AVANTAGES

- ✓ Représente la pertinence graduelle, plus réaliste.
- ✓ Supporte les combinaisons complexes de termes (ET/OU/NON).
- ✓ Plus flexible que le modèle booléen strict.

#### INCONVÉNIENTS

- Ne repose pas sur une base probabiliste.
- Les résultats peuvent dépendre de la pondération choisie (TF-IDF, fréquence...).
- Peut être moins précis que le modèle vectoriel pour les requêtes longues.