

- Rappel index inversé :

"terme1"	"CACM-1"	tf.idf
	...	
	"CACM-xxx"	tf.idf
"terme2"	"CACM-2"	tf.idf
	...	
	"CACM-zzz"	tf.idf
...		

- Vecteur requête : v_q

"terme_x"	tf.idf
...	
"terme_y"	tf.idf

- Norme de $v_q \rightarrow nv_q$

- Traitement de requête (point III. a) :

- On boucle sur les termes de vq un par un

- Exemple : terme_x

- Récupérer la ligne de terme_x dans l'index inversé

et l'affecter dans

ligne_trav :

"CACM-2"	tf.idf
...	
"CACM-zzz"	tf.idf

"terme_x"	tf.idf
...	
"terme_y"	tf.idf

- Mettre à jour ligne_trav en multipliant par le poids de terme_x dans vq.

ligne_trav :

"CACM-2"	tf.idf * tf.idf
...	
"CACM-zzz"	tf.idf * tf.idf

- INSERER ligne_trav dans resultat_partiel (cf. page suivante).

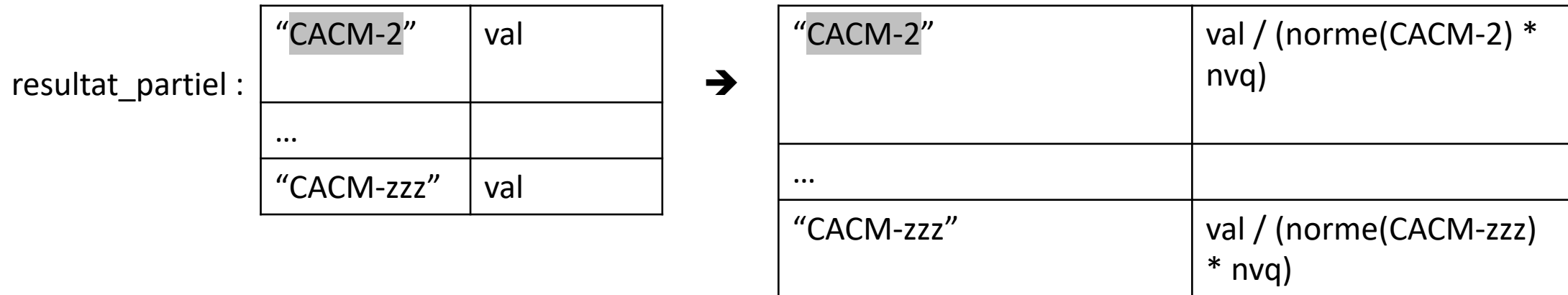
resultat_partiel :

"CACM-2"	tf.idf * tf.idf
...	
"CACM-zzz"	tf.idf * tf.idf

Note : resultat_partiel stocke au fur et à mesure des termes de vq le produit scalaire entre requête et docs qui ont des éléments en commun.

- Explication sur l'insertion de ligne_trav:
 - On passe en revue tous les éléments de ligne_trav
 - pour une clé "CACM-x"
 - Si "CACM-x" existe comme clé dans resultat_partiel
 - On met à jour resultat_partiel["CACM-x"] += ligne_trav["CACM-x"]
 - Sinon
 - On crée la clé "CACM-x" dans resultat_partiel , en initialisant par :
resultat_partiel["CACM-x"] = ligne_trav["CACM-x"]

- Traitement de requête (point III. b) :
 - On finalise le calcul du cosinus en divisant le resultat_partiel après passage en revue de tous les termes par les norms des documents et de la requête



- Point IV : Ensuite il ne reste qu'à trier resultat_partiel par ordre de valeur décroissante, ne garder que les n premiers et afficher les identifiants de docs

• Traitement de requête (point III. a) requête avec 2 termes:

• On boucle sur les termes de vq un par un

• terme_x

• Récupérer la ligne de terme_x dans l'index inversé

et l'affecter dans

ligne_trav :

"CACM-2"	tf.idf
...	
"CACM-zzz"	tf.idf

Hypothèse :
CACM-1 pas dans la ligne de terme_x

• Mettre à jour ligne_trav en multipliant par le poids de terme_x dans vq.

ligne_trav :

"CACM-2"	tf.idf * tf.idf
...	
"CACM-zzz"	tf.idf * tf.idf

• INSERER ligne_trav dans resultat_partiel (cf. page suivante).

resultat_partiel :

"CACM-2"	tf.idf * tf.idf
...	
"CACM-zzz"	tf.idf * tf.idf

"terme_x"	tf.idf
"terme_y"	tf.idf

- Traitement de requête (point III. a) requête avec 2 termes:

- On boucle sur les termes de vq un par un

- $terme_y$

- Récupérer la ligne de $terme_x$ dans l'index inversé et l'affecter dans

ligne_trav :

"CACM-1"	tf.idf
...	
"CACM-zzz"	tf.idf

CACM-1 EST DANS la ligne de $terme_x$

- Mettre à jour `ligne_trav` en multipliant par le poids de $terme_x$ dans vq .

ligne_trav :

"CACM-1"	tf.idf * tf.idf
...	
"CACM-zzz"	tf.idf * tf.idf

Rappel :
CACM-1 pas dans la ligne de $terme_x$

- INSERER `ligne_trav` dans `resultat_partiel` (cf. page suivante).

resultat_partiel :

"CACM-1"	= tf.idf * tf.idf
"CACM-2"	inchangé
...	
"CACM-zzz"	+= tf.idf * tf.idf