Capitole Speciale de Informatică

Curs 3: Scoruri de potrivire a documentelor. Calculul greutăţii termenilor. Modelul de spaţiu vectorial

11 octombrie 2018

Modelul boolean de extragere a informaţiilor are o limitare importantă: nu poate calcula un scor de importanţă care să indice cât de mult răspunde un document unei cereri de informare. Este de dorit ca pentru fiecare pereche (d, q) în care

- d este un document, şi
- q este o cerere de informare

să putem calcula un scor $scor(d, q) \in \mathbb{R}$ care măsoară gradul de potrivire al lui d ca răspuns la cererea q: cu cât documentul d răspunde mai bine cererii q, cu atât $scor(d, q)$ este mai mare.

De ce este util calculul unui scor de potrivire?

- ▶ În general, căutarea documentelor care se potrivesc cu o cerere q produce o listă uriaşă de rezultate. Identificarea documentelor care răspund cel mai bine cererii q este dificilă.
- ▶ Scorurile de potrivire permit ordonarea şi afişarea rezultatelor în ordine descrescătoare a importanţei lor ca răspunsuri.

În acest curs sunt prezentate 3 tehnici folosite în implementarea sistemelor de extragere a informaţiilor, care permit calculul unui scor de importanţă:

1. Indexarea metadatelor caracteristice unui document. Acest proces produce indecşi parametrici şi indecşi de zonă, care permit (1) căutari bazate pe metadate (de ex., titlu sau limbaj folosit); şi (2) o metodă simplă de calcul al unui scor de importanţă.
2. Calculul unei măsuri de importanţă, numită **greutate**, pentru fiecare termen din dicţionar.

3. Reprezentarea documentelor ca vectori M-dimensionali de greutăţi, unde $V = \{t_1, \ldots, t_M\}$ este dicţionarul de termeni al colecţiei de documente. Fiecare document d este reprezentat de vectorul $(g_1, g_2, \ldots, g_M)$ unde g_i este scorul de importanţă al lui t_i în documentul d. Această reprezentare uşurează calculul scorurilor de potrivire $scor(d, q)$.

1 Scoruri de potrivire bazate pe metadate

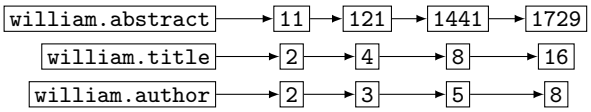
Majoritatea documentelor în format electronic au, pe lângă conţinutul text, şi **metadate**. Metadatele sunt forme specifice de date despre un document, şi pot fi de 2 feluri:

Câmpuri. Fiecare câmp are un nume şi o mulţime finită de valori posibile. Exemple de nume de câmpuri de metadate de document sunt: data creării documentului; limba în care este scris documentul; etc.

Pentru fiecare câmp se construieşte un **index parametric** care permite găsirea rapidă a tuturor documentelor pentru care câmpul respectiv are o anumită valoare.

Zone. O zonă are un nume şi o valoare care este un text oarecare oricât de mare. Exemple de nume de zone sunt: titlu; autor; rezumat (sau abstract); etc. Indexarea zonelor se poate face în 2 feluri:

1. Se construieşte un index inversat separat pentru fiecare zonă, de exemplu: un index pentru abstract-uri, altul pentru titluri, şi altul pentru autori. De pildă, listele de indecşi inversaţi pentru termenul `william` în zonele din metadate ar putea arăta astfel:



2. Se extind postările pentru termenii din zone cu specificări ale zonelor în care apar. De exemplu:

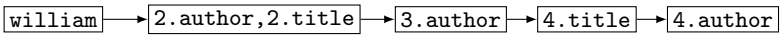


Figura 1 ilustrează interfaţa unui sistem de extragere a informaţiilor cu indecşi parametrici şi de zone.

1.1 Scoruri bazate pe greutatea zonelor

Listele de postări pentru termenii din zonele de metadate permit o tehnică simplă de definire a unui scor de potrivire pentru zone în intervalul $[0, 1]$:

1. Presupunem că există ℓ zone de metadate distincte $z_1, \ldots, z_\ell$. Fixăm o pondere de importanţă $g_i \in [0, 1]$ pentru fiecare zonă z_i. Suma ponderilor zonelor trebuie să fie 1, adică $\sum_{i=1}^{\ell} g_i = 1$.

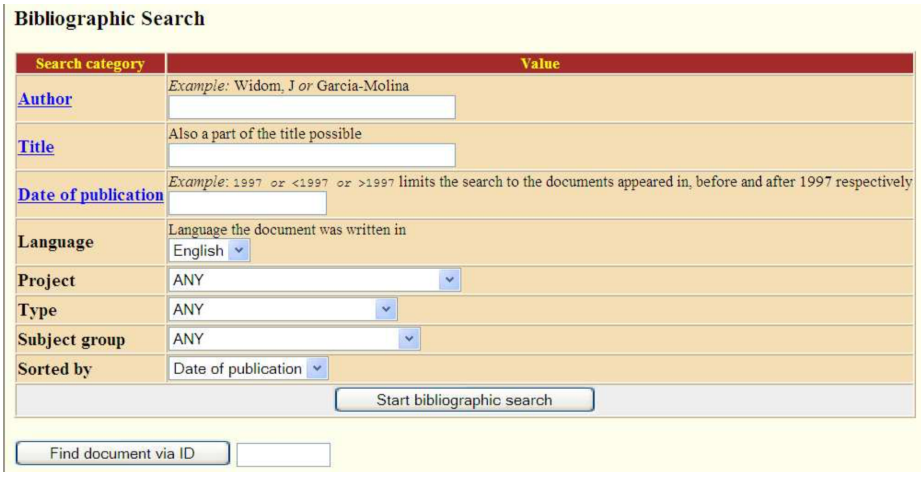


Figure 1: Căutare bazată pe indecşi parametrici şi indecşi de zonă.

2. Fie $s_i \in \{0, 1\}$ valoarea booleană de potrivire a zonei z_i a documentului d cu cererea q. Scorul (sau gradul) de potrivire dintre zonele lui d şi q este dat de formula

$$scor(d, q) := \sum_{i=1}^{\ell} g_i \cdot s_i$$

De exemplu, dacă avem o colecţie D de documente cu trei zone: z_1 pentru `author`, z_2 pentru `title` şi z_3 pentru `body`, şi fixăm ponderile $g_1 = 0.2$, $g_2 = 0.3$ şi $g_3 = 0.5$ atunci:

- Scorul de potrivire $scor(d, \texttt{shakespeare})$ pentru un document în care termenul `shakespeare` apare în zonele pentru `title` şi pentru `body` este

$$0.2 \cdot s_1 + 0.3 \cdot s_2 + 0.5 \cdot s_3 = 0.2 \cdot 0 + 0.3 \cdot 1 + 0.5 \cdot 1 = 0.8$$

Această tehnică de calcul al unui scor de potrivire pentru zone se numeşte **extragere booleană gradată a informaţiilor**.

Listele de indecşi inversaţi pentru zone de metadate pot fi folosite pentru calculul direct al scorurilor de potrivire. De exemplu, dacă q_1 şi q_2 sunt două cereri, iar

- p_1 este lista de postări care indică documentele şi zonele ce se potrivesc cu cererea q_1 (metoda 2 de la pagina 2)
- p_2 este lista de postări care indică documentele şi zonele ce se potrivesc cu cererea q_2 (metoda 2 de la pagina 2)

atunci putem calcula direct un tablou $scoruri[\,]$ care, pentru fiecare document d al cărui identificator $docId$ apare în p_1 şi p_2, reţine în $scoruri[docId]$ scorul

de potrivire dintre d şi cererea q_1 `AND` q_2. Algoritmul de calcul este ilustrat în figura 2 în care presupunem că

1. colecţia are N documente, şi fiecare document are ℓ zone cu ponderile $g[1], \ldots, g[\ell]$
2. metoda auxiliară ScorDeZonă(p_1, p_2) este apelată cu două postări p_1, p_2 pentru acelaşi document d; p_1 indică zonele lui d care se potrivesc cu cererea q_1; p_2 indică zonele lui d care se potrivesc cu cererea q_2; metoda întoarce valoarea $\sum_{i=1}^{\ell} g_i \cdot s_{1,i} \cdot s_{2,i}$ unde $s_{j,i}$ este 1 dacă zona i a lui d se potriveşte cu cererea q_j, şi 0 în caz contrar.

```
ScoruriDeZonă(q1, q2)
 1 float scoruri[N] = [0]
 2 constant g[ℓ]
 3 p1 := postings(q1)
 4 p2 := postings(q2)
 5 // scoruri este un tablou cu o intrare iniţializată pentru fiecare docID
 6 // p1, p2 sunt iniţializaţi să se refere la începuturile listelor de postări
 7 // Presupunem că g[ ] este iniţializat cu ponderile corespunzătoare de zonă
 8 while p1 ≠ Nil and p2 ≠ Nil do
 9       if =docID(p1) = docID(p2) then
10          scoruri(docID(p1)) :=ScorDeZonă(p1, p2, g)
11          p1 := next(p1)
12          p2 := next(p2)
13       else if docID(p1) < docID(p2) then
14          p1 := next(p1)
15       else p2 := next(p2)
16 return scoruri
```

Figure 2: Calculul scorurilor ponderate de zonă pentru cererea q_1 `AND` q_2.

1.2 Învăţarea scorurilor

Determinarea ponderilor de zonă $g_1, \ldots, g_\ell$ se face adesea cu un algoritm de învăţare care funcţionează astfel:

- Primeşte un set T de documente clasificate deja de către nişte operatori umani. Setul T se numeşte set de **exemple de antrenare**.
- Determină ponderile $g_1, \ldots, g_\ell$ care minimizează o măsură de eroare a scorurilor calculate pentru exemplele de antrenare.
- Ponderile calculate se vor folosi pentru calculul scorurilor unor documente noi.

Exemplu ilustrat

Fie D o colecţie de documente cu două zone: $\mathtt{T}$ pentru titlu şi $\mathtt{C}$ pentru conţinutul propriu-zis (corpul) documentului. În acest caz, avem de fixat o valoare $g \in [0,1]$ cu care calculăm scorul de potrivire

$$scor(d,q) = g \cdot s_{\mathtt{T}}(d,q) + (1-g) \cdot s_{\mathtt{C}}(d,q).$$

În formula de mai sus avem $s_{\mathtt{T}}(d,q) = 1$ dacă zona $\mathtt{T}$ a lui d se potriveşte cu cererea q, şi $s_{\mathtt{T}}(d,q) = 0$ în caz contrar. Deasemenea, $s_{\mathtt{C}}(d,q) = 1$ dacă zona $\mathtt{C}$ a lui d se potriveşte cu cererea q, şi $s_{\mathtt{C}}(d,q) = 0$ în caz contrar.

În acest caz, o mulţime de exemple de antrenare este o mulţime finită de tripleţi $\Phi_j = (d_j, q_j, r(d_j,q_j))$ formate din un document d_j, o cerere q_j şi o decizie de relevanţă: $r(d_j,q_j) = 1$ dacă documentul d_j se consideră relevant pentru cererea q, şi $r(d_j,q_j) = 0$ în caz contrar.

Se consideră că **eroarea de calcul al scorului** pentru exemplul de antrenare Φ_j este $\epsilon(g,\Phi_j) := (r(d_j,q_j) - scor(d_j,q_j))^2$, iar **eroarea totală de calcul al scorurilor de antrenare** este

$$\sum_{j=1}^{M} \epsilon(g,\Phi_j) \quad \text{dacă } T = \{\Phi_1, \Phi_2, \ldots, \Phi_M\}.$$

Se doreşte să se aleagă valoarea $g \in [0,1]$ pentru care eroarea totală ia valoarea minimă. Fie

- $n_{ij\mathtt{r}}$ numărul exemplelor de antrenare $(d_j, q_j, r(d_j,q_j))$ cu $s_{\mathtt{T}}(d_j,q_j) = i$ $s_{\mathtt{C}}(d_j,q_j) = j$ şi $r(d_j,q_j) = 1$.
- $n_{ij\mathtt{n}}$ numărul exemplelor de antrenare $(d_j, q_j, r(d_j,q_j))$ cu $s_{\mathtt{T}}(d_j,q_j) = i$ $s_{\mathtt{C}}(d_j,q_j) = j$ şi $r(d_j,q_j) = 0$.

Rezultă că eroarea totală este o expresie polinomială de gradul 2 în g:

$$\sum_{j=1}^{M} \epsilon(g,\Phi_j) = (n_{01\mathtt{r}} + n_{10\mathtt{n}}) \cdot g^2 + (n_{10\mathtt{r}} + n_{01\mathtt{n}})(1-g)^2 + n_{00\mathtt{r}} + n_{11\mathtt{n}}$$

Deci, avem de rezolvat o problemă de minimizare pătratică. Valoarea lui g pentru care derivata acestei expresii este 0 este

$$\frac{n_{00\mathtt{r}} + n_{11\mathtt{n}}}{n_{00\mathtt{r}} + n_{00\mathtt{n}} + n_{11\mathtt{r}} + n_{11\mathtt{n}}} \in [0,1]$$

Rezultă că aceasta este valoarea optimă a lui g pe care o calculează algoritmul de învăţare.

2 Calculul frecvenţei şi al greutăţii termenilor

În modelul boolean de extragere a informaţiilor contează doar dacă un termen apare sau nu într-un document. O variantă îmbunătăţită ar fi să considerăm că

un document sau zonă este cu atât mai relevant(ă) pentru o cerere exprimată de un termen t cu cât t apare mai frecvent în documentul sau zona respectivă.

Prin urmare, vom calcula **frecvenţa unui termen în un document**:

$\mathrm{tf}_{t,d}$:= numărul de apariţii al termenului t în documentul d.

> **Observaţie:** calculul frecvenţelor termenilor ne permite să interpretăm fiecare document ca pe o colecţie de termeni, în care numărul de apariţii al termenilor contează, dar ordinea în care apar termenii nu contează. De exemplu, documentul cu conţinutul "Ion este mai înalt decât George" este identic în această interpretare cu documentul cu conţinutul "George este mai înalt decât Ion."

Nu toţi termenii sunt la fel de importanţi pentru a descrie conţinutul specific unui document. De exemplu, este firesc ca termenul `auto` să apară foarte des într-o colecţie de documente referitoare la industria auto, şi să nu transmită informaţii noi. Pentru a reduce relevanţa termenilor care apar frecvent în colecţii şi care nu au relevanţă mare pentru descrierea conţinutului documentului, se calculează valorile următoare:

- **frecvenţa de colecţie a unui termen**:

  cf_t := numărul totel de aparc tii al lui t în colecţia de documente D.

- **frecvenţa de document a unui termen**:

  df_t := numărul totel de documente din colecţia D în care apare t.

În general, aceste valori se comportă diferit. De exemplu, frecvenţele de colecţie şi de document ale termenilor `try` şi `insurance` au valorile ilustrate mai jos în o colecţie de documente de ştiri ale agenţiei Reuters:

termen	cf	df
`try`	10422	8760
`insurance`	10440	3997

Pentru a reduce relevanţa termenilor care apar în multe documente, se defineşte factorul numit **frecvenţa inversă de document** a lui t:

$$\mathrm{idf}_t := \log \frac{N}{\mathrm{df}_t}$$

unde N este numărul total de documente în colecţia D. Se observă că idf_t ia valori mari pentru termeni ce apar în puţine documente din colecţie, şi ia valori mici pentru termeni ce apar în multe documente din colecţie. Figura 3 ilustrează acest comportament.

În final, se defineşte **greutatea** $\text{tf-idf}_{t,d}$ a unui termen t în un document d:

$$\text{tf-idf}_{t,d} := \mathrm{tf}_{t,d} \cdot \mathrm{idf}_t.$$

Se observă că $\text{tf-idf}_{t,d}$ atribuie o greutate

- mare când t apare des în un număr mic de documente,
- redusă când t apare rar în d, sau apare în multe documente,
- foarte redusă când t apare în majoritatea documentelor din colecţie.

termen	df_t	idf_t
car	18165	1.65
auto	6723	2.08
insurance	19241	1.62
best	25235	1.5

Figure 3: Exemplu de valori df şi idf pentru termeni din o colecţie de 806791 documente a agenţiei de ştiri Reuters.

3 Modelul de spaţiu vectorial pentru documente

Calculul scorurilor de potrivire al documentelor cu cereri de informare se simplifică dacă reprezentăm fiecare document d ca pe un vector de dimensiune M

$$\vec{V}(d) := (\text{tf-idf}_{t_1,d}, \ldots, \text{tf-idf}_{t_M,d})$$

unde $V = \{t_1, \ldots, t_M\}$ este dicţionarul de termeni ai colecţiei de documente. Documentele devin vectori într-un spaţiu M-dimensional care are o axă pentru fiecare termen, iar componenta i a vectorului are valoarea $\text{tf-idf}_{t_i,d}$ care indică relevanţa lui t_i în conţintul lui d.

3.1 Similaritatea documentelor

Modul standard de calcul al similarităţii a 2 documente este dat de similaritatea cosinusoidală a vectorilor $\vec{V}(d_1)$ şi $\vec{V}(d_2)$:

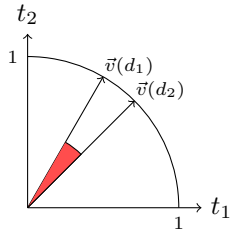
$$\text{sim}(d_1, d_2) := \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)| \cdot |\vec{V}(d_2)|}.$$

Reamintim că, dacă $\vec{x} = (x_1, \ldots, x_M)$ şi $\vec{y} = (y_1, \ldots, y_M)$ sunt doi vectori de aceeaşi dimensiune M, atunci

- **produsul lor scalar** este $\vec{x} \cdot \vec{y} := \sum_{i=1}^{M} x_i y_i$
- **lungimea** vectorului $\vec{x}$ este $|\vec{x}| := \sqrt{\sum_{i=1}^{M} x_i^2}$
- $\dfrac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$ este valoarea cosinusului unghiului θ dintre vectorii $\vec{x}$ şi $\vec{y}$.

De exemplu, dacă $\vec{v}(d_1) = \vec{V}(d_1)/|\vec{V}(d_1)|$ şi $\vec{v}(d_2) = \vec{V}(d_2)/|\vec{V}(d_2)|$ atunci

- $\vec{v}(d_1)$ şi $\vec{v}(d_2)$ au lungimea 1, iar cosinusul unghiului dintre ei este produsul scalar $\vec{v}(d_1) \cdot \vec{v}(d_2)$.



- Cosinusul unghiului roşu dintre $\vec{v}(d_1)$ şi $\vec{v}(d_2)$ reprezintă similaritatea dintre d_1 şi d_2. Se observă că

$$-1 \leq \text{sim}(d_1, d_2) \leq 1.$$

Dacă $D = \{d_i \mid 1 \leq i \leq N\}$ este o colecţie de documente şi vrem să găsim documentele din D cele mai similare cu un document dat d, putem proceda astfel: calculăm toate produsele scalare $\vec{v}(d_i) \cdot \vec{v}(d)$, unde

$$\vec{v}(d_i) = \frac{\vec{V}(d_i)}{|\vec{V}(d_i)|} \quad \text{şi} \quad \vec{v}(d) = \frac{\vec{V}(d)}{|\vec{V}(d)|}$$

şi se returnează documentele d_i pentru care produsele scalare iau valorile cele mai mari.

O colecţie cu N documente şi M termeni poate fi reprezentată ca o matrice $M \times N$ în care rândurile reprezintă M termeni iar coloanele reprezintă documentele din colecţie.

3.2 Reprezentarea vectorială a cererilor

Considerăm expresii booleene de forma

$$q = t_{i_1} \ t_{i_2} \ \ldots \ t_{i_n}$$

ca cereri de informare pentru documente care conţin termenii $t_{i_1}, \ldots, t_{i_n}$. Presupunem că extragerea informaţiilor se face din o colecţie de N documente cu un vocabular de M termeni. În acest caz, cererea q se reprezintă cu vectorul

$$\vec{v}(q) := (v_1, \ldots, v_M) \quad \text{unde } v_i = \begin{cases} \frac{1}{\sqrt{n}} & \text{dacă } i \in \{i_1, \ldots, i_n\}, \\ 0 & \text{în caz contrar.} \end{cases}$$

Scorul de potrivire al cererii q cu un document q este

$$\vec{v}(q) \cdot \vec{v}(d) \quad \text{unde } \vec{v}(d) = \frac{\vec{V}(d)}{|\vec{V}(d)|}.$$

Exemplu ilustrat

Presupunem că D este o colecţie fictivă de $N = 10^6$ documente în care apar doar termenii `auto`, `best`, `car` şi `insurance`. Deasemenea, considerăm că cererea q este

```
best car insurance
```

şi că avem valorile statistice indicate în tabelul de mai jos:

termen	cerere				document			produs
	tf	df	idf	$w_{t,q}$	tf	wf	$w_{t,d}$	
auto	0	5000	2.3	0	1	1	0.41	0
best	1	50000	1.3	1.3	0	0	0	0
car	1	10000	2.0	2.0	1	1	0.41	0.82
insurance	1	1000	3.0	3.0	2	2	0.82	2.46

În acest exemplu greutatea $w_{t,q}$ a unui termen t în cererea q este valoarea idf (care este 0 pentru termenii care nu apar în cerere, ca de exeplu `auto`). În documente, se consideră că greutatea $w_{t,d}$ a unui termen t în un document d este valoarea normalizată a lui $tf_{t,d}$. Deci

$$\vec{v}(q) = (0, 1.3, 2.0, 3.0)$$
$$\vec{v}(d) = (0.41, 0.0.41, 0.82)$$

deci scorul de potrivire al lui q cu d este $\vec{v}(q) \cdot \vec{v}(d) = 0 + 0 + 0.82 + 2.46 = 3.28$.

3.3 Găsirea primelor K documente care se potrivesc cel mai bine cu o cerere

De obicei, un sistem de extragere a informaţiilor are o colecţie de documente reprezentate ca vectori $\{\vec{v}(d_i) \mid 1 \leq i \leq N\}$, o cerere q reprezentată ca un vector $\vec{v}(q)$, şi i se cere să găsească primele K documente din colecţie care se potrivesc cel mai bine cu cererea q. Pseudocodul din figura 4 este pentru algoritmul de bază care calculează scorurile de potrivire a documentelor din D cu cererea q. În acest pseudocod, $lungime[d] = \sqrt{\sum_{i=1}^{M} wf_{t_i,d}}$ este lungimea euclidiană a vectorului $\vec{v}(d) = (wf_{t_1,d}, \ldots, wf_{t_M,d})$ iar în pasul 7, $wf_{t,d}$ este de obicei fie $tf_{t,d}$ sau tf-$idf_{t,d}$.

Sunt două posibilităţi de memorare a listei de postări pentru un termen t:

1. $t \longrightarrow \boxed{d_1 \mid tf_{t,d_1} \mid wf_{t,d_1}} \longrightarrow \cdots \longrightarrow \boxed{d_n \mid tf_{t,d_n} \mid wf_{t,d_n}} \longrightarrow \cdots$

   adică să se reţină valoarea lui wf_{t,d_i} (un număr `float` sau `double`) împreună cu postarea lui d_i pentru termenul t.

2. $t \longrightarrow \boxed{N/df_t} \longrightarrow \boxed{d_1 \mid tf_{t,d_1}} \longrightarrow \cdots \longrightarrow \boxed{d_n \mid tf_{t,d_n}} \longrightarrow \cdots$

```
ScorCosinusoidal(q)
 1 float scoruri[N] = [0]
 2 for fiecare d din colecţia D do
 3     iniţializează lungime[d]
 4 for fiecare termen t din cererea q do
 5     calculează w_{t,q} şi obţine lista de postări a lui t
 6     for fiecare pereche (d, tf_{t,d}) din lista de postări a lui t do
 7         scoruri[d] += wf_{t,d} · w_{t,q}
 8 for fiecare document d ∈ D do
 9     scoruri[d] := scoruri[d]/lungime[d]
10 return primele K componente din scoruri[ ]
```

Figure 4: Algoritmul de bază pentru calculul scorurilor de potrivire în spaţiul vectorial de documente.

> Prima intrare din lista de postări este specială: reţine valoarea lui N/df_t (un `float` sau `double`). Această codificare a listei de postări ocupă mai puţin spaţiu şi permite (1) calculul lui $\mathrm{idf}_t := \log \dfrac{N}{\mathrm{df}_t}$, şi (2) calculul valorilor $\text{tf-idf}_{t,d_i} := \mathrm{idf}_t \cdot \mathrm{tf}_{t,d_i}$.

Presupunând că $q = t_{i_1}\ t_{i_2}\ \ldots\ t_{i_n}$, se observă că, la pasul 9, algoritmul memorează în $scoruri[d]$ valoarea

$$\frac{\sum_{k=1}^{n} \mathrm{wf}_{t_{i_k},d} \cdot \mathrm{w}_{t_{i_k},q}}{lungime[d]}$$

Valoarea lui $scoruri[d]$ se calcuează incremental: după ce termenul t_{i_j} este ales din cererea q în bucla **for** (paşii 4–7), fiecare element de tablou $scoruri[d]$ va reţine valoarea $\sum_{k=1}^{j} \mathrm{wf}_{t_{i_k},d} \cdot \mathrm{w}_{t_{i_k},q}$. Acest calcul incremental al scorurilor de potrivire se numeşte *gradare după termeni* (engl. *term-at-a-time scoring*) sau **acumulare**, iar elementele tabloului $scoruri[\,]$ se numesc **acumulatori**.

Bibliografie

1. Capitolul 6 din
   Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze: *An Introduction to Information Retrieval*. Ediţie online (c) 2009 Cambridge UP.
   `http://nlp.stanford.edu/IR-book/pdf/irbookonlinereading.pdf`