Capitole Speciale de Informatică

Curs 1: Extragerea informaţiilor. Modelul boolean şi modelul boolean extins

27 septembrie 2018

Extragerea informaţiilor (engl. *Information Retrieval, IR*) constă în găsirea şi extragerea de material (de obicei, documente) nestructurat sau semistructurat (de obicei, care conţine text) care satisface o **necesitate de informare** din colecţii mari (care, de obicei, sunt stocate pe calculatoare).

- **Datele nestructurate** sunt date fără o structură precisă care să poată fi procesată uşor cu un program pe calculator.
  - Exemple de date structurate: o bază de date relaţională cu informaţii despre angajaţii unei companii.
  - Majoritatea datelor textuale au o structură lingvistică latentă: titlu, secţiuni, paragrafe, etc. Astfel de surse de informaţii se numesc *documente semistructurate*.
- Necesităţile de informare sunt comunicate unui sistem de extragere a informaţiilor prin intermediul unei **cereri** (engl. **query**) formulate de către utilizator.
- Un document este **relevant** pentru necesitatea de informare a unui utilizator dacă utilozarorul respecitv consideră că documentul respectiv conţine informaţii valoroase pentru nevoia lui de informare.

Exemple tipice de probleme rezolvate de către sistemele de extragere a informaţiilor:

- **Căutare semistructurată**, de exemplu, găsirea documentelor al căror titlu conţine "Java" şi al căror conţinut (corp) conţine "threading."
- **Clustering**, adică împărţirea unei colecţii mari de documente în grupuri (clustere) care conţin documente cu conţinut asemănător.
- **Clasificare**: dacă se dă (1) o mulţime de tematici ce corespund unor necesităţi de informare, sau (2) alte categorii de text (de exemplu, pentru cititori de anumite vârste), să se decidă la ce tematică sau categorie aparţine fiecare document.

  - Adesea se bazează pe algoritmi de învăţare ce pornesc de la o mulţime de documente clasificate manual, şi se speră că algoritmul este suficient de bun încât să înveţe să clasifice corect documentele noi.

- **Căutare al-hoc** a documentelor din o colecţie care sunt relevante pentru orice necesitate de informare comunicată sistemului printr-o cerere iniţiată de către utilizator.

Documentele de extragere a informaţiilor pot fi clasificate după mărimea colecţiei de documente pe care o analizează:

- Sistemele de **căutare web** analizează miliarde de documente stocate pe milioane de calculatoare. Un astfel de sistem este optimizat
  - să analizeze un volum cât mai mare de documente existente pe web (*crawling*)
  - să proceseze caracteristici specifice documentelor web: hiperlink-uri, tentative de manipulare a scorului de importanţă, etc.

- Sistemele de **căutare pentru un mediu de afaceri, instituţie** sau **un anumit domeniu**.
  - Colecţiile de documente analizate pot fi: documente interne ale unei corporaţii; o bază de date de patente; articole de cercetare despre biochimie; etc.
  - De obicei, aceste colecţii sunt stocate în sisteme centralizate de fişiere, iar căutarea de informaţii în ele de face cu ajutorul câtorva calculatoare dedicate.

- Sisteme de **extragere a informaţiilor personale** precum Spotlight (Mac OS X). Instant Search (Windows Vista), sau programe de email cu capabilităţi de clasificare automată a mesajelor, filtrare spam:
  - analizează o gamă largă de tipuri de documente stocate pe un calculator personal
  - sunt rapide, ocupă puţină memorie, şi nu afectează semnificativ experienţa de lucru a utilizatorului cu calculatorul

1 Modelul boolean de extragere a informaţiilor

Se presupune dată o colecţie de documente D. Are următoarele caracteristici:

1. Cererea de informare este o expresie booleană B care descrie ce termeni dorim să apară sau nu în documentele pe care le căutăm în colecţia D. Se pot folosi conectorii logici `AND`, `OR` şi `NOT`:

   B `::=` $\langle$`termen`$\rangle$ | `NOT` B | B `AND` B | B `OR` B

2. Se consideră că un document este doar o mulţime de termeni.

3. Documentele din colecţie sunt **indexate** înainte de a efectua cereri de informare. Indexarea se face astfel: se citesc toate documentele, se extrag toţi termenii care apar în ele, şi se indexează, adică se reţin toate documentele în care apare termenul respectiv.

   - Mulţimea tuturor termenilor din colecţia D se numeşte **vocabular** sau **dicţionar** şi o notăm V.
   - Rezultatele indexării pot fi reţinute în o **matrice de incidenţă** $V \times D$ în care elementul de la poziţia (t, d) este 1 dacă termenul t apare în documentul d, şi 0 în caz contrar.

   Un exemplu de matrice de incidenţă pentru termenii ce apar în câteva din operele lui Shakespeare este ilustrat mai jos:

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Anthony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
...						

   Rândul din matrice corespunzător unui termen t este un şir de n biţi, unde n este numărul de documente din d. De exemplu, şirul de biţi pentru Brutus este `110100`, pentru Caesar este `110111`, şi pentru Calpurnia este `010000`.

4. Permite căutarea ad-hoc folosind expresii booleene cu termeni ca să comunice necesităţile de informare ale utilizatorilor.

Necesitatea de informare pentru documentele care conţin termenii `Brutus` şi `Caesar` dar nu conţin `Calpurnia`, poate fi comunicată cu expresia booleană

```
Brutus AND Caesar AND NOT Calpurnia
```

iar răspunsul se obţine prin operaţii booleene pe vectori de biţi:

```
110100 AND 110111 AND NOT 010000 =
110100 AND 110111 AND 101111 = 100100
```

Documentele returnate ca răspuns la cererea utilizatorului sunt primul şi al patrulea, adică Anthony and Cleopatra şi Hamlet, fiindcă rezultatul `100100` are primul şi al patrulea bit cu valoarea `1`.

1.1 Statistici de estimare a eficacităţii unui sistem de IR

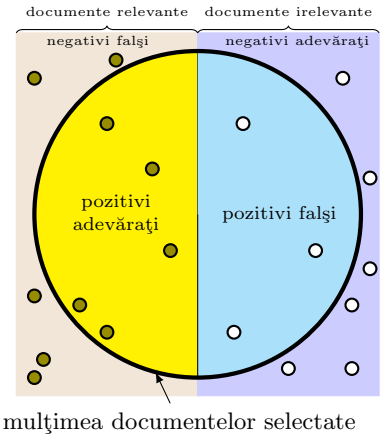
Eficacitatea unui sistem de extragere a informaţiior se estimează folosind următoarele măsurători statistice:

- **Precizia**: ce procent din rezultatele returnate de sistem sunt documente relevante pentru necesitatea de informare a utilizatorului?
- **Reamintirea** (engl. **recall**): ce procent din documentele relevante din colecţie sunt printre rezultatele returnate de sistem ca răspuns la necesitatea de informare a utilizatorului (adică pozitivi adevăraţi)?

Exemplu: Fie S este un sistem de extragere a informaţiilor din o colecţie D de $N = 19$ documente. Presupunem că un utilizator comunică cererea Q sistemului S şi că:

- ▶ D conţine $N_r = 10$ documente relevante pentru Q
- ▶ Sistemul S returnează $N_s = 8$ documente ca răspunsuri la cerere. N_s este numărul documentelor selectate de către S din D ca răspunsuri.
- ▶ 5 din cele 8 răspunsuri sunt documente relevante pentru Q (sau *pozitivi adevăraţi*). Celelalte 3 răspunsuri sunt documente irelevante pentru Q (sau *pozitivi falşi*).

Situaţia din acest exemplu este ilustrată în diagrama de mai jos.



● : document relevant
○ : document irelevant

$$\text{precizie} = \frac{\text{pozitivi adevăraţi}}{N_s} = \frac{5}{8} = 0.625$$

$$\text{reamintire} = \frac{\text{pozitivi adevăraţi}}{N_r} = \frac{5}{10} = 0.5$$

Precizia măsoară **exactitatea** sau **calitatea** răspunsurilor.

Reamintirea măsoară **completitudinea** sau sau **cantitatea** răspunsurilor.

1.2 Liste de indecşi inversaţi

Sunt o alternativă de reprezentare a rezultatului indexării termenilor care, de obicei, ocupă mult mai puţin spaţiu de memorie decât matricea de incidenţă.

De exemplu, dacă D este o colecţie de $N \approx 10^6$ documente alcătuite fiecare din $\approx 10^3$ termeni, iar fiecare termen ocupă în medie 6 octeţi, atunci toată colecţia D are o mărime de ≈ 6 GB. Putem estima că D conţine $N \approx 5 \cdot 10^5$ termeni distincţi.

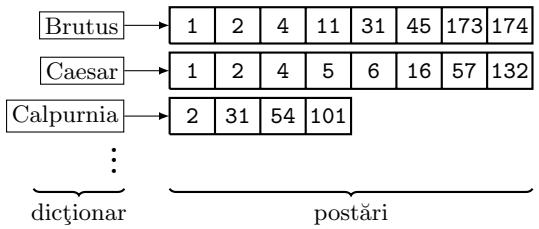
veste rea: matricea de incidenţă pentru colecţia D de documente diin exemplu ocupă $5 \cdot 10^5 \times 10^6 \approx 0.5 \cdot 10^{12}$ biţi: nu încape în memoria RAM a unui calculator obişnuit

veste bună: matricea conţine $\approx 10^6$ elemente care sunt 1. Rezultă că $\approx 99.8\%$ din elementele matricii sunt 0 $\Rightarrow$ este mai economic să reţinem în liste poziţiile la care apare 1 în matrice.

Aceste liste se numesc **liste de indecşi inversaţi** sau **liste de postări**.

- pentru fiecare termen $t \in V$ reţinem lista de identificatori de documente (docID) în care apare t.
- identificatorii de documente sunt reţinuţi în ordine crescătoare în liste.

De acum încolo vom presupune că fiecare document are un număr serial unic care este o valoare întreagă pozitivă. Acest număr se numeşte **identificator de document**. De exemplu, listele de indecşi inversaţi pentru termenii din operele lui Shakespeare pot arăta astfel:



Listele de indecşi inversaţi se construiesc în 4 paşi:

1. Se colectează documentele care urmează să fie indexate $\Rightarrow$ o colecţie de N documente D cu identificatorii $d_1, \ldots, d_N$.
2. Se transformă textele documentelor în liste de token-uri. Acest pas se numeşte **teeknizare**.

   De exemplu, tokenizarea textului

   > "Friends, Romans, countrymen. So let it be with Caesar."

   este [Friends] [Romans] [countrymen] [So] [let] [it] [be] [with] [Caesar]
3. Se face o preprocesare lingvistică a token-urilor $\Rightarrow$ o listă de termeni normalizaţi numiţi **termeni**. Termenii sunt elementele care se indexează. De exemplu, lista de termeni pentru textul ilustrat mai sus este

   [friend] [roman] [countryman] [so] ...
4. Indexarea propriu-zisă. După pasul 3 se vor cunoaşte

- mulţimea V a tuturor termenilor care apar în textele din colecţia D. Această mulţime de termeni V se numeşte **dicţionar** şi se sortează crescător.
- lista de identificatori de documente în care apare fiecare termen t. Aceste liste se sortează crescător.
- valori statistice despre textele documentelor. De exemplu, **frecvenţa de document** a fiecărui termen t, care reprezintă numărul de documente în care apare termenul t.

1.3 Procesarea cererilor booleene

Procesarea unei cereri booleene B_1 `AND` B_2 produce o listă de postări care satisfac cererile B_1 şi B_2. Această listă se determină în felul următor:

1. Se determină listele de postări p_1 pentru B_1, şi p_2 pentru B_2.
2. Se intersectează listele L_1 şi L_2. Intersecţia se poate face rapid, în timp $O(m+n)$ unde m este lungimea lui p_1 şi n este lungimea lui p_2, cu algoritmul următor:

```
INTERSECT(p1, p2)
1 raspuns := ⟨⟩
2 while p1 ≠ Nil and p2 ≠ Nil do
3       if docID(p1) = docID(p2)
4          adaugă docID(p1) la raspuns
5          p1 := next(p1)
6          p2 := next(p2)
7       else if docID(p1) < docID(p2)
8               p1 := next(p1)
9       else p2 := next(p2)
10 return raspuns
```

Procesările celorlalte tipuri de cerei booleene

`NOT` B şi B_1 `OR` B_2

se pot defini în mod asemănător.

2 Extensii ale modelului boolean

Modelul boolean de extragere a informaţiilor este prea limitat fiindcă (1) fiecare document este interpretat doar ca o mulţime de termeni distinţi, şi (2) nu poate calcula un scor de importanţă al documentelor, care să indice care documente sunt răspunsuri mai importante la o cerere de căutare.

Câteva din aceste limitări au fost eliminate de un **model boolean extins** de extragere a informaţiilor. Acest model are un **operator de proximitate** cu

care putem preciza cât de aproape trebuie să fie 2 termeni în documentele care ne interesează.

Westlaw (`http://www.westlaw.com/`) este un sistem comercial de căutare într-o colecţie de zeci de terabytes de documente despre legi si legislaţie. Westlaw are un model boolean extins de extragere a informaţiilor. Exemple interesante de cereri de căutare mai sofisticate cu Westlaw sunt:

- `"trade secret" /s disclos! /s prevent /s employe!`
  căutarea de informaţii despre legi de prevenire a obţinerii de secrete comerciale de către angajaţi care au lucrat anterior la o companie competitoare.
- `disab! /p access! /s work-site work-place (employment /3 place)`
  Cerinţe pentru persoane cu handicap care îşi caută un loc de muncă.