6. Extragerea de cunostinţe din web

Puncte importante :

1. ***Numărarea dinamică a mulţimilor de articole***: Căutarea de mulţimi ***interesante*** de articole într-un spaţiu mult prea mare pentru a se putea lua în considerare fiecare pereche de articole.
2. ***"Cărţi şi autori"***: Intrigantul experiment al lui Sergey Brin de extragere de date relaţionale din web.

6.1 Găsirea mulţimilor de articole neobişnuite

Problema este de a găsi mulţimi de cuvinte care apar "neobişnuit de des" împreună pe web, e.g. "New" şi "York" sau {Ducesa, de, York}.

- "Neobişnuit de des" poate fi definit în diverse moduri pentru a încorpora ideea că numărul de documente web conţinând mulţimea de cuvinte este mult mai mare decât cel aşteptat în cazul în care cuvintele ar fi fost alese la întamplare, fiecare cuvânt cu probabilitatea sa de apariţie într-un document.
- Un mod adecvat este ***entropia per cuvânt din mulţime***. Formal, ***interesul*** unei mulţimi de cuvinte S este:

$$\frac{\log_2\left(\frac{prob(S)}{\prod_{w \text{ in } S} prob(\mathrm{w})}\right)}{|S|}$$

De notat că împărţim la dimensiunea lui S pentru a evita "efectul Bonferroni", în care sunt atât de multe mulţimi de o dimensiune dată încât unele, din motive probabilistice, par a fi corelate.

- Exemplu: Daca ***a***, ***b*** şi ***c*** (cuvinte) apar fiecare în 1% din toate documentele şi S={a, b, c} apar în 0.1% din documente, interesul lui S este $(\log_2(0.001/(0.01 \times 0.01 \times 0.01))/3 = \log_2(1000)/3$ adică aproximativ 3.3.
- Problema tehnică: interesul nu este monoton sau "închis în jos" în modul de la produse cu larg suport. Asta înseamnă că putem avea o mulţime S cu o valoare mare a interesului şi totuşi unele sau chiar toate submulţimile sale stricte să nu fie interesante. Prin contrast, dacă S are suport larg, atunci toate submulţimile sale au cel puţin acelaşi suport.
- Problemă tehnică: Cu mai mult de 10^8 cuvinte diferite apărând pe web nu este posibil nici măcar să considerăm toate perechile de cuvinte.

6.2 Motorul DICE

DICE (dynamic itemset counting engine) vizitează repetat paginile web într-un mod de tip "round-robin". De fiecare dată numără apariţiile anumitor mulţimi de cuvinte şi ale fiecărui cuvânt din aceste mulţimi. Numarul de mulţimi numărate este suficient de mic încât contorii lor încap în memoria centrală.
Din când în când, să spunem la fiecare 5000 de pagini, DICE îşi reconsideră mulţimile pentru care numără. Înlătură acele mulţimi care au cel mai mic interes şi le înlocuieşte cu alte mulţimi.
Alegerea noilor mulţimi se bazează pe ***proprietatea muchiei grelei (heavy edge property)*** care este o observaţie justificată experimental că acele cuvinte care apar în mulţimi cu interes ridicat au probabilitatea mai mare să apară în alte mulţimi cu interes ridicat. Astfel, când selectează noi mulţimi pentru a începe numărarea, DICE este direcţionat în favoarea cuvintelor care apar deja în mulţimi cu interes ridicat. Totuşi, el nu se bazează exclusiv pe aceste cuvinte altfel nu ar putea niciodată să găsească mulţimi cu interes ridicat compuse din multele cuvinte pe care nu le-a considerat niciodată. Unele (dar nu toate) din construcţiile pe care le utilizează DICE pentru crearea noilor mulţimi sunt:

1. Două cuvinte aleatoare. Aceasta este singura regulă independentă de aserţiunea muchiei grele şi ajuta noi cuvinte să ajungă în mulţime.
2. Un cuvânt dintr-una din mulţimile interesante şi un cuvânt aleator.

3. Două cuvinte din două mulţimi interesante diferite.
4. Reuniunea a două mulţimi interesante a căror intersecţie are dimensiunea 2 sau mai mult.
5. {a, b, c} dacă toate mulţimile {a, b}, {a, c} şi {b, c} sunt găsite ca fiind interesante.

Bineînţeles, în general sunt mult prea multe opţiuni de a aplica cele de mai sus în toate modurile posibile astfel încât se utilizează o selecţie aleatoare a opţiunilor dând o anumită şansă fiecăreia dintre ele.

6.3 Cărţi şi autori

Ideea principală este de a căuta pe web fapte de un anumit tip, de genul celor care ar putea forma o relaţie de genul *Cărţi(titlu, autor)*. Procesarea este sugerată de Fig. 13.

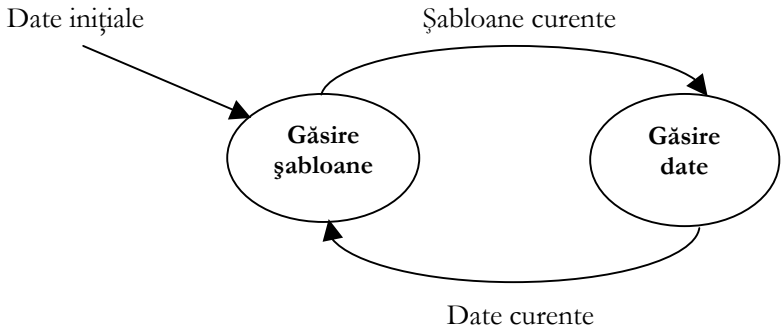

Figura 13: Extragerea relaţiilor din web

1. Se porneşte de la un eşantion al tuplelor care se doresc găsite. În exemplul discutat în lucrarea lui Brin au fost folosite cinci exemple de titluri de cărţi şi autori ai acestora.
2. Fiind dată o mulţime de exemple cunoscute, se caută pagini unde apar aceste date pe web. Dacă se găseşte un şablon care identifică un număr de tupluri cunoscute şi este suficient de specific încât e puţin probabil să identifice prea mult, atunci se acceptă acest şablon.
3. Fiind dată o mulţime de şabloane acceptate, se caută date care satisfac aceste şabloane şi se adaugă la mulţimea datelor cunoscute.
4. Se repetă paşii (2) şi (3) de un număr de ori. În exemplul citat au fost utilizate patru ciclări care au dus la 15,000 de tuple; aprox. 95% au fost perechi adevărare titlu-autor.

6.4 Cum arată un şablon?

Noţiunea constă în cinci elemente:

1. *Ordinea*; i.e. daca titlul apare în text înaintea autorului sau vice-versa. Într-un caz general, în care tuplele au mai mult de 2 componente, ordinea va fi dată de permutarea componentelor.
2. *Prefixul adresei web (URL).*
3. *Prefixul* textului, care apare înaintea primului dintre titlu şi autor
4. *Mijlocul*: text care apare între cele două elemente de date.
5. *Sufixul* textului care urmează după al doilea dintre cele două elemente de date. Atât prefixul cât şi sufixul au fost limitate la 10 caractere.

Exemplul 6.1 : Un şablon posibil poate consta din următoarele:

1. Ordinea: titlul şi apoi autorul.
2. Prefixul URL: `www.stanford.edu/class`
3. Prefixul, mijlocul şi sufixul de forma următoare:

```
<LI><I>titlu</I> de autor<P>
```

   Aici prefixul este `<LI><I>`, mijlocul este `</I> de` (inclusiv spaţiul după "de") şi sufixul este `<P>`. Titlul este orice apare între prefix şi mijloc; autorul este orice apare între mijloc şi sufix.

Intuiţia pentru care acest şablon poate fi bun este că există probabil o mulţime de liste cu referinţe bibliografice în paginile cursurilor de la Stanford. □

Pentru a se focaliza pe şabloanele care pot fi corecte, Brin a utilizat o serie de constrângeri asupra şabloanelor, după cum urmează:

- Definim ***specificitatea*** unui şablon ca fiind produsul lungimilor prefixului, mijlocului, sufixului şi prefixului URL. În mare, specificitatea măsoară cât de posibil este să găsim date care corespund şablonului; cu cât specificitatea este mai mare, cu atât ne aşteptăm la mai puţine apariţii ale acestuia în date.
- Apoi şablonul trebuie să îndeplinească două condiţii pentru a fi acceptat:
  1. Trebuie să fie cel puţin 2 elemente de date cunoscute care apar conform acelui şablon.
  2. Produsul specificităţii şablonului cu numărul de apariţii de date conform acestuia trebuie să depăşească un anumit prag T (nespecificat).

6.5 Apariţii ale datelor

O apariţie a unui tuplu este asociată cu un şablon după care acestea apar; i.e., acelasi titlu şi autor pot să apară după diferite şabloane. Astfel, o apariţie a datelor constă în:

1. Un anumit titlu şi autor.
2. Adresa Internet completa (URL) şi nu doar prefixul ca în cazul şablonului.
3. Ordinea, prefixul, mijlocul şi sufixul şablonului după care au apărut titlul şi autorul respectiv.

6.6 Găsirea aparţiilor pornind de la datele cunoscute

Dacă avem câteva perechi autor-titlu cunoscute, primul pas în găsirea de noi şabloane este căutarea pe web pentru a vedea unde apar aceste titluri şi autori. Să presupunem că există un index al web-ului astfel încât dându-se un cuvânt putem gasi toate (legăturile către) paginile conţinând acel cuvânt. Metoda utilizată este esenţial a-priori:

1. Găsim (legături către) toate paginile conţinând oricare dintre autorii cunoscuţi. Cum numele autorilor constau în general din 2 cuvinte, se utilizează indexul pentru fiecare dintre prenume şi nume şi se verifica dacă apariţiile sunt consecutive în document
2. Găsim (legături către) toate paginile conţinând oricare dintre titlurile cunoscute. Se porneşte prin găsirea paginilor conţinând toate cuvintele titlului şi apoi verificând că aceste cuvinte apar în ordine în pagină.
3. Se intersectează mulţimile de pagini care au un autor şi un titlu în ele. Doar pentru aceste pagini este nevoie să se facă operaţia de căutare pentru găsirea şabloanelor în care se găseşte o pereche cunoscută autor-titlu. Pentru prefix şi sufix se iau cele 10 caractere alăturate acestora sau mai puţine dacă nu exista 10.

6.7 Construcţia şabloanelor din apariţiile de date

1. Se grupează apariţiile de date după ordinea şi mijlocul lor. De exemplu, un grup din acest "group-by" poate corespunde ordinii "titlu-apoi-autor" şi mijlocului "`</I> de "`.
2. Pentru fiecare grup se găseşte cel mai lung prefix, sufix şi prefix URL comun.
3. Dacă testul de specificitate pentru acest şablon este îndeplinit, se acceptă şablonul.
4. Dacă testul de specificitate *nu* este îndeplinit, se încearcă spargerea grupului în doua prin extinderea lungimii prefixului URL cu un caracter şi apoi se repetă pasul (2). Dacă este imposibil să spargem grupul (pentru că există doar un URL) atunci am eşuat în a produce un nou şablon din acel grup.

Exemplul 6.6 : Să presupunem că grupul conţine trei URL-uri:

```
www.stanford.edu/class/cs345/index.html
www.stanford.edu/class/cs145/index.html
www.stanford.edu/class/cs340/readings.html
```

Prefixul comun este `www.stanford.edu/class/cs`. Dacă trebuie să spargem grupul, atunci următorul caracter, 3 sau 1, sparge grupul în două, cu acele apariţii ale datelor din prima pagină (pot fi multe astfel de apariţii) mergând într-un prim grup şi apariţiile din celelalte două pagini în celălalt.

6.8 Găsirea apariţiilor pornind de la şabloane

1. Se găsesc toate URL-urile care se potrivesc cu prefixul URL al cel puţin unui şablon.
2. Pentru fiecare astfel de pagină se parcurge textul folosind o expresie regulată construită din prefixul, mijlocul şi sufixul şablonului.
3. Se extrage din fiecare potrivire titlul şi autorul, dupa ordinea specificată în şablon.