Algoritmul coloniei de furnici pentru dezambiguizarea nesupervizata a textelor (*Ant colony algorithm*)

Bibliografie

- **Ant Colony Algorithm for the Unsupervised Word Sense Disambiguation of Texts: Comparison and Evaluation** (D. Schwab, J. Goulian, A. Tchechmedjiev, H. Blanchon) in "Processing of COLING 2012", Mumbai, paginile 2389-2404

- Lucrarea se refera la *knowledge-based unsupervised word sense disambiguation* (este la granita dintre cele doua)

- Procesul de invatare este nesupervizat dar algoritmului i se dau cunostinte (foloseste Lesk extins, deci cunostinte date de WordNet)
- Masura Lesk locala este propagata intr-un intreg text. In general un algoritm global este o metoda care permite propagarea unei masuri locale in cadrul unui intreg text pentru a atribui un sens fiecarui cuvant al textului.

Analiza se face la nivelul intregului text; fereastra de context se inlocuieste cu tot textul (algoritm global)

Notatii:

- w_i = cuvant ambiguu
- m = numar de cuvinte in intregul text
- $w_{i,j}$= cel mai adecvat sens al cuvantului w_i fiind dat contextul.
- $d(w_{i,j})$= definitia unui sens j al cuvantului i

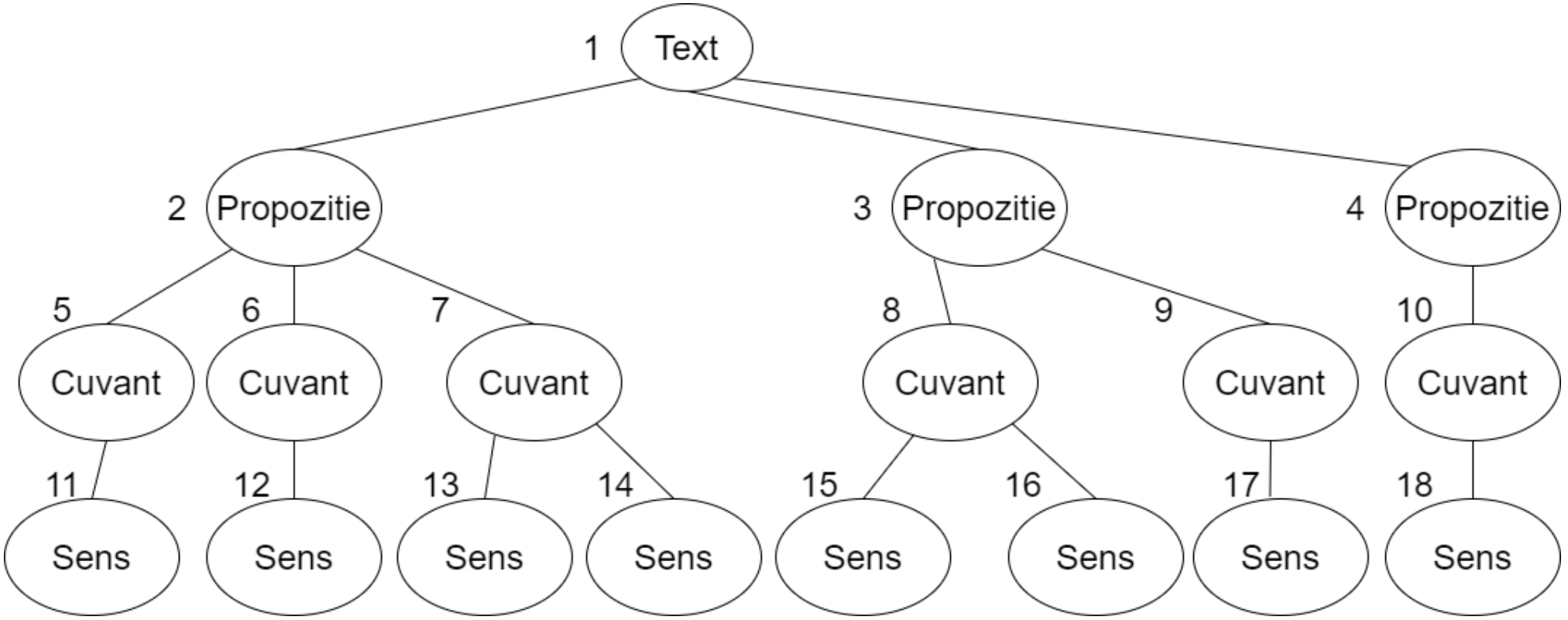
- Spatiul de cautare corespunde tuturor combinatiilor posibile ale tuturor sensurilor cuvintelor textului care se proceseaza.
- Cu C se noteaza o configuratie a problemei.
- Reprezentarea unei configuratii C a problemei consta intr-un vector de intregi astfel incat j = C[i] este sensul j selectat pentru cuvantul w_i

Furnici artificiale

- Au fost prima data folosite pentru a rezolva “Traveling Salesman Problem” (Dorigo si Gambardella, 1997)
- Mediul este, de obicei, reprezentat printr-un graf, in interiorul caruia furnici virtuale exploreaza dare de feromon (drumuri) depozitat de acele furnici
- Avantaje:
  - O buna alternativa pentru rezolvarea problemelor modelate prin grafuri
  - Adaptabilitate mare la schimbarile de mediu (medii cu grad mare de schimbare)

- Fiecare sens posibil al unui cuvant este asociat unui cuib (nod frunza)
- Cuiburile produc furnici care se misca in graf
- Furnicile cara un miros (matrice) care contine cuvintele din definitia sensului corespunzator cuibului mama al furnicii
- Tipuri de noduri pentru o furnica:
  - Cuibul mama (unde s-a nascut furnica)
  - Cuib dusman (corespunde unui alt sens al aceluiasi cuvant)
  - Potential cuib prieten (reprezentat de orice alt cuib)
  - Un nod simplu (orice nod care nu este cuib)
- Fiecarui nod simplu, ii este asociat un vector de miros, de lungime fixata, care initial este vid.

- Exemplu de graf pentru reprezentare:



Radacina arborelui corespunzatoare grafului e intreg textul.
De exemplu, cuvantul corespunzator nodului 7 este un cuvant polisemantic.

- Miscarea furnicilor depinde de:
  - Scorurile acordate de algoritmul local
  - Prezenta energiei
  - Trecerea altor furnici (atunci cand trec, furnicile lasa o dara de feromon, care se evapora cu timpul)
  - Vectorii de miros (*odour*) ale nodurilor (furnicile depoziteaza o parte din mirosul lor in nodurile prin care trec)
- Atunci cand o furnica ajunge in cuibul unui alt termen (i.e. un nod care corespunde unui sens) poate decide: sa continue explorarea sau sa construiasca un pod intre cele doua cuiburi si sa urmeze acest pod pentru a ajunge acasa.

- Podurile se comporta ca niste muchii obisnuite, numai ca, atunci cand concentratia de feromon corespunzatoare lor atinge valoarea 0, podul se prabuseste.
- Cu cat mai apropiate sunt sensurile corespunzatoare cuiburilor, cu atat mai mult un pod intre ele va contribui la intarirea lor reciproca si la partajarea de resurse intre ele
- Podurile dintre sensuri mai indepartate vor avea tendinta sa se prabuseasca
- In acest fel se creeaza drumuri de interpretare ("*interpretative paths*"), adica posibile interpretari ale textului, in urma unui comportament emergent. Tot in acest fel este eliminata necesitatea folosirii unui graf intreg (care include toate legaturile posibile intre sensuri)

- Executia algoritmului consta intr-un numar de cicluri potential infinit. Dupa fiecare ciclu se observa starea mediului (“*state of the environment*”)
- Un ciclu se compune din urmatorii pasi:
  - Se elimina furnicile moarte si podurile care nu mai au feromon
  - Corespunzator fiecarui cuib se produce o furnica
  - Pentru fiecare furnica se determina modul in care se afla (cauta energie sau se intoarce); o facem sa se miste; se creeaza un pod de interpretare, daca este posibil (“*interpretative bridge*”)
  - Se actualizeaza mediul – *update the environment* (nivelul de energie al nodurilor, feromon si vectori de miros – *odour vectors*)

Notatii principale pentru algoritm:

F_A – cuib care corespunde sensului A

f_A – furnica nascuta in cuibul F_A

V(X) – vector de miros asociat lui X (furnica sau nod)

E(X) – energia lui X (unde X poate fi furnica sau nod)

$Eval_f(N)$ – evaluarea unui nod N de catre o furnica f

$Eval_f(A)$ – evaluarea unei muchii A (cantitatea de feromon) de catre o furnica f

$\varphi_{(t/c)}(A)$ – cantitatea de feromon pe o muchie A la momentul t sau ciclul c

Parametri pentru algoritm:

Notatie	**Descriere**
E_a	Energia luata de o furnica atunci cand ajunge la un nod
E_{max}	Cantitatea maxima de energie pe care o poate cara o furnica
δ	Rata de evaporare a feromonului intre doua cicluri
E_0	Cantitatea initiala de energie din fiecare nod
ω	Durata de viata a furnicii
L_v	Lungimea vectorului de mirosuri
δ_v	Proportia de componente de miros depozitate de catre o furnica intr-un nod oarecare
c_{ac}	Numarul de cicluri al simularii

1. Nasterea furnicilor, moartea si modelul de energie

Initial se atribuie o cantitate fixa de energie, E_o, fiecarui nod al mediului. La inceputul fiecarui ciclu, fiecare nod cuib N are posibilitatea sa produca o furnica A, folosind o unitate de energie, cu probabilitatea $P(N_A)$. In literatura, aceasta probabilitate este definita ca fiind functia:

$$P(N_A)=\frac{\arctan(E(N))}{\pi}+0.5$$

unde E(N) este energia nodului N

Atunci cand este creata, o furnica are o durata a vietii alcatuita din **ω** cicluri. Atunci cand viata ei atinge valoarea 0, furnica este distrusa (stearsa) la inceputul urmatorului ciclu, iar energia ei este depozitata in nodul unde a murit. In felul asta se asigura pastrarea **echilibrului global de energie al sistemului**, ceea ce joaca un rol fundamental in convergenta catre o solutie.

2. Miscarile furnicilor

Miscarea furnicilor este aleatoare, dar influentata de mediu.

Atunci cand o furnica se afla intr-un nod, ea atribuie o probabilitate de tranzitie muchiilor care conduc spre toate nodurile vecine. Probabilitatea de a trece printr-o muchie A_j pentru a ajunge intr-un nod N_i este:

$$P(N_i, A_j)=\frac{Eval_f(N_i,A_j)}{\sum_{k=1,l=1}^{k=n,l=m} Eval_f(N_k,A_l)}$$

unde $Eval_f(N,A)=Eval_f(N)+Eval_f(A)$ este functia de evaluare a unui nod N atunci cand se ajunge in el venind de-a lungul muchiei A

O furnica nou-nascuta incepe sa caute mancare in felul urmator:

a) Este atrasa de nodurile care au cea mai mare cantitate de energie:

$Eval_f(N)=\frac{E(N)}{\sum_{i=0}^{m} E(N_i)}$

b) Evita sa mearga de-a lungul muchiilor cu mult feromon:

$Eval_f(A)=1-\varphi_t(A)$

unde $\varphi_t(A)$ este cantitatea de feromon a muchiei A la momentul t

pentru a favoriza o explorare mai ampla a spatiului de cautare

Furnica colecteaza atata energie (mancare) cat este posibil, pana cand decide sa o aduca inapoi acasa – si intra in “*return mode*” cu probabilitatea:

$$P(return)=\frac{E(f)}{E_{max}}$$

unde E(f) este energia pe care o are furnica f la momentul current

iar E_{max} este cantitatea maxima de energie pe care o poate transporta o furnica

Observatie: *Atunci cand o furnica f isi atinge capacitatea maxima de transport, probabilitatea ca ea sa se intoarca este 1.*

Cand decide sa se intoarca, se deplaseaza urmand statistic acele muchii care contin cel mai mult feromon:

$Eval_f(A)=\varphi_t(A)$

$Eval_f(A)$ reprezinta evaluarea muchiei A de catre furnica f (cantitate de feromon)

$\varphi_t(A)$ reprezinta cantitatea de feromon a muchiei A la momentul t

De asemenea, muchiile trebuie sa conduca spre noduri cu un miros (odour) apropiat de al ei:

$$Eval_f(N)=\frac{ExtLesk(V(N),V(f_A))}{\sum_{i=1}^{k} ExtLesk(V(N_i),V(f_A))}$$

V(N) – vector de miros asociat nodului N

$V(f_A)$ – vector de miros asociat furnicii nascute in nodul F_A

ExtLesk() – algoritmul Lesk extins

Observatie: Algoritmul Lesk extins se aplica asupra a doi vectori de miros.

3. Crearea si stergerea podurilor; tipuri de poduri

Atunci cand o furnica ajunge intr-un nod adiacent unui potential cuib prieten (care corespunde unui sens al cuvantului) trebuie sa decida intre a urma oricare dintre caile posibile sau a merge la acel nod cuib. Deci avem un caz particular al algoritmului de alegere a drumului de catre furnica, cazul cu:

$Eval_f(A)=0$ (i.e. feromonul muchiei este ignorat)

Singura diferenta este ca, daca furnica alege sa mearga la potentialul cuib prieten, se construieste un pod intre acest cuib si cuibul parinte al furnicii. Furnica parcurge acest pod pentru a se intoarce acasa.

Podurile se comporta ca si celelalte muchii, cu exceptia situatiei in care concentratia de feromon corespunzatoare lor atinge valoarea 0. In acest caz, podul se prabuseste si este inlaturat.

4. Modelul de feromon

Atunci cand se misca in graf, furnicile lasa dare de feromon de-a lungul muchiilor pe care le parcurg.

Furnicile au doua tipuri de comportament:

- cauta sa acumuleze energie
- vor sa se intoarca la cuibul mama

Miscarea furnicilor in graf este influentata de densitatea feromonului corespunzator fiecarei muchii: **ele prefera sa evite muchiile cu mult feromon atunci cand cauta energie** (hrana) si **le urmeaza** pe acestea atunci **cand vor sa transporte energia inapoi la cuibul mama**

Atunci cand se deplaseaza de-a lungul unei muchii A, furnicile lasa o dara depozitand o cantitate de feromon $\theta \in \mathbb{R}^+$ a.i.

$\varphi_{t+1}(A)=\varphi_t(A) + \theta$

In plus, , corespunzator fiecarui ciclu, exista o evaporare liniara a feromonului (care penalizeaza drumurile putin frecventate):

$\varphi_{t+1}(A)=\varphi_t(A) * (1 - \delta)$

unde δ reprezinta rata de evaporare a feromonului

5. Miros

Definitie: *Mirosul unui cuib este un vector de valori numerice asociat sensului respectiv (sensul pentru care s-a creat cuibul).* Acest vector numeric al sensului se determina in felul urmator:

- Se aplica Lesk extins (Banerjee &Pedersen, 2002), usor modifcat, astfel incat fiecare cuvant continut in oricare dintre definitii (glose) este indexat printr-un unic numar intreg. Modificarea consta in faptul ca suprapunerile de siruri de cuvinte nu primesc un scor mai mare (“la patrat”), cuvintele individuale fiind tratate in mod individual (suprapunere de tip “*bag of words*”). Aceasta se face pentru a micsora complexitatea de la O(m*n) la O(m), m>n, unde m si n sunt lungimile celor doua definitii care se compara. Prin aceasta indexare, intreaga definitie primeste un scor reprezentat printr-un vector.

Exemplu:

Definitia: “Some kind of evergreen tree”

S-a determinat ca:

“same” este indexat prin 123

“kind” este indexat prin 14

“evergreen” este indexat prin 34

“tree” este indexat prin 90

atunci reprezentarea indexata a intregii definitii este:

{14,34,90,123} (Observatie: vectorul este sortat)

Toate furnicile **nascute in acelasi cuib** au acelasi vector al mirosului. Atunci cand o furnica ajunge intr-un nod oarecare N, ea depune in acel nod unele dintre componentele vectorului sau de miros (urmand o distributie uniforma). Acestea vor fi adaugate la, sau vor inlocui componenta existenta a vectorului nodului, V(N).

Mirosul nodurilor cuib nu se modifica niciodata.

Acest mecanism permite furnicilor sa regaseasca drumul inapoi la nodul lor in cuib. Cu cat un nod este mai apropiat de un cuib dat, cu atat mai multe furnici ale acelui cuib au trecut prin acel nod si au depus componente de miros.

Prin urmare, mirosul unui nod va reflecta vecinatatea cuibului respectiv si va permite furnicilor sa gaseasca calea calculand scorul intre mirosul lor (cel al cuibului parinte) si mirosul nodurilor inconjuratoare. In urma acestui calcul, vor alege sa se deplaseze la nodul cu scorul cel mai mare.

Acest proces permite si existenta erorilor (de exemplu, o furnica ajunge in alt cuib decat cel propriu). Procesul este insa benefic pentru ca le determina pe furnici sa construiasca mai multe poduri.

Evaluare globala

- La sfarsitul fiecarui ciclu, se construieste configuratia curenta a problemei pe baza formei grafului: pentru fiecare cuvant, se alege sensul corespunzand cuibului avand cea mai mare cantitate de energie.
- In continuare, se calculeaza scorul global al configuratiei curente:
  - Scorul unui sens selectat al unui cuvant poate fi exprimat ca fiind suma scorurilor locale intre acel sens si sensurile selectate pentru toate celelate cuvinte ale unui context (o propozitie)
  - Pentru scorul global al configuratiei se aduna scorurile pentru toate sensurile selectate corespunzator cuvintelor textului:

$$Scor(C) = \sum_{i=1}^{m} \sum_{j=1}^{m} ExtLesk\left(w_{i,C[i]}, w_{j,C[j]}\right)$$

j=C[i] este sensul selectat j pentru cuvantul w_i

Observatie: Aici se vede ca nu se lucreaza cu o fereastra de context ci cu tot textul.

- Complexitate: $O(m^2)$, $\text{unde } m = \text{numarul}$ de cuvinte din text.

- In timpul executiei algoritmului se pastreaza configuratia care are scorul cel mai mare si care va fi folosita la sfarsit pentru a genera solutia.

Valorile uzuale ale parametrilor algoritmului (articolul original, 2012):

Notatie	Descriere	Valoare
E_a	Energia luata de o furnica atunci cand ajunge la un nod	1 – 30
E_{max}	Cantitatea maxima de energie pe care o poate cara o furnica	1 – 60
δ	Rata de evaporare a feromonului intre doua cicluri	0.0 – 1.0
E_0	Cantitatea initiala de energie din fiecare nod	5 – 60
ω	Durata de viata a furnicii	1 – 30 (cicluri)
L_v	Lungimea vectorului de mirosuri	20 – 200
δ_v	Proportia de componente de miros depozitate de catre o furnica intr-un nod oarecare	0 – 100%
c_{ac}	Numarul de cicluri al simularii	1 – 500

Valorile parametrilor nu pot fi determinate in mod analitic si au fost evaluate in mod experimental.

Experimentul care s-a facut a constat in adnotarea a 2269 cuvinte cu unul dintre sensurile lor posibile date de WordNet. Gradul mediu de polisemie intalnit a fost de 6.19 (S-a folosit corpusul de la SemEval 2007).

In urma simularilor si a testelor s-au gasit urmatoarele valori optime ale parametrilor:

$\omega = 25, E_a$=16, E_{max}=56, E_0=30, δ_v=0.9, δ=0.9, L_v=100

Testele din anul 2013 au condus la urmatoarele valori ale parametrilor:

- Pentru limba engleza

$\omega = 26, E_a$=14, E_{max}=3, E_0=34, δ_v=0.9775, δ=0.3577, L_v=25

- Pentru limba franceza

$\omega = 19, E_a$=9, E_{max}=3, E_0=32, δ_v=0.9775, δ=0.3577, L_v=25

Acuratete obtinuta (F1) pe datele SemEval-2007: **76.41%**

Algoritmul a fost superior celui mai bun “*state of the art*” sistem nesupervizat si celui mai slab sistem supervizat.

Caracteristicile algoritmului:

- Dezambiguizare globala (intregul text este context de dezambiguizare)
- Este de tip *knowledge based unsupervised* (se dau cunostinte din WordNet; procesul de invatare este de tip nesupervizat)