ASAP – Sistem avansat de evaluare a participanților la un chat

Mihai Dascălu
Facultatea de Automatică şi Calculatoare
Universitatea Politehnica Bucureşti
mikedascalu@yahoo.com

Erol-Valeriu Chioaşcă
Facultatea de Automatică şi Calculatoare
Universitatea Politehnica Bucureşti
erolvaleriu@yahoo.com

Ştefan Trăuşan-Matu
Facultatea de Automatică şi Calculatoare
Universitatea Politehnica Bucureşti
trausan@cs.pub.ro

REZUMAT
Lucrarea propune o metodă de evaluare a celui mai competent participant din cadrul unui mediu colaborativ de tip *chat*. Articolul descrie, de asemenea, programe *software* implementate care uşurează vizualizarea rezultatelor analizelor. Aceste programe ajută atât profesorii cât şi studenții în îmbunătățirea metodelor de învățare şi în evaluarea participanților oferind un feedback pe baza *chat*-ului.

Cuvinte cheie
Învățare colaborativă, chat, rețele sociale, LSA

INTRODUCERE

În ultimii ani, o dată cu dezvoltarea şi răspândirea calculatoarelor, au apărut noi medii de comunicare, gen *chat* (mesagerie instantanee), care au determinat schimbarea ideii de învățare colaborativă ajutată de calculator, ea devenind o alternativă în a sprijini învățământul clasic [8]. Din această perspectivă, a apărut şi necesitatea unor instrumente de asistare a interacțiunii, în contextul sistemelor pentru învățământ fiind foarte utilă o evaluare automată a contribuției participanților, luând în considerare o serie de factori aplicați atât asupra numărului şi structurării replicilor interschimbate, cât şi asupra conținutului semantic al acestora. Lucrarea de față prezintă un astfel de sistem, denumit ASAP (An Advanced System for Assessing Chat Participants – sistem avansat pentru evaluarea partcipanților la un chat).

În analiza propusă în lucrare se porneşte de la succesiunea replicilor unui chat şi se construieşte rețeaua socială (vezi http://www.orgnet.com/sna.html, precum şi [5]) care poate fi reprezentată vizual prin graful participanților (figura 1), precum şi graful replicilor interschimbate, util pentru vizualizarea ierarhiei/înşiruirii replicilor. Rețeaua socială este un graf în care nodurile sunt participanții la discuție iar arcele sunt replicile schimbate.

Graful replicilor reprezintă o reprezentare ierarhică a succesiunii acestora, bazată pe legăturile explicite marcate în transcriptul *XML*-ul exportat din sistemul *Polyphony* [6] sau ConcertChat [3].

Lucrarea continuă cu prezentarea factorilor folosiți de sistem în analiza contribuției participanților la chat, în scopul evaluării lor. Următoarea secțiune trece în revistă câteva detalii de implementare. Ultima secțiune înainte de concluzii prezintă un modul folosit în adnotarea sesiunilor salvate de chat, în scopul obținerii unui corpus de referință pentru evaluare.

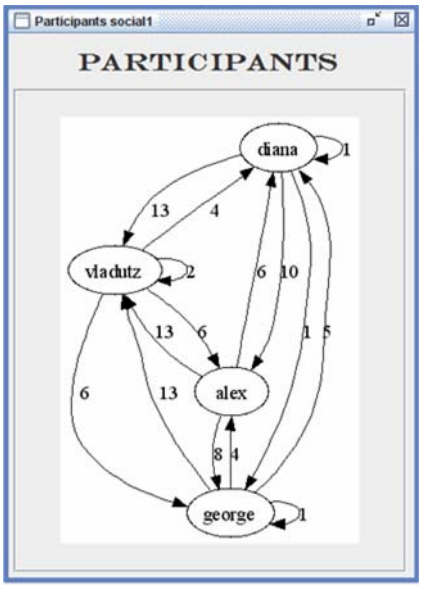


Figura 1 Graful participanților

FACTORI ÎN ANALIZĂ

A. Numărul de caractere

Un factor simplu, dar de multe ori relevant în descoperirea persoanei celei mai competente dintr-un *chat*, este numărul de caractere scrise de aceea persoană. Bineînteles că este de dorit să nu se urmărească neapărat cantitatea informației scrise, ci mai degrabă calitatea, însă pentru a o determina pe cea din urmă este utilă şi cunoşterea acestui factor.

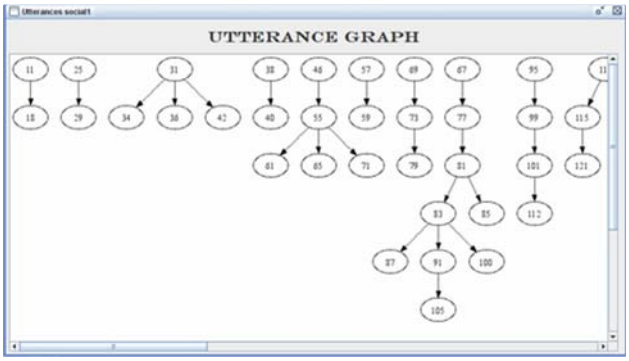


Figura 2 Graful replicilor

B. Numărul de caractere per intervenție

În acelaşi spirit în care cantitatea joacă un rol important în analiza discursului unui participant, numărul de caractere per intervenție influențează nota finală deoarece determină eficiența medie a intervenției. Eficiența medie este privită

în această lucrare ca o balanță între lungimea intervenției şi consistența informațiilor pe care le cuprinde.

C. Grad

Plecând de la transcriptul *chat*-ului, este generat un graf în concordanță cu numărul de replici interschimbate de participanți. Din punctul de vedere a teoriei grafurilor, sunt luate în considerare două măsuri:

- I-grad (in-degree) – numărul arcelor care intră într-un nod din graf;
- E-grad (out-degree) – numărul arcelor care ies dintr-un nod din graf.

Numărul de arce (numărul de replici interchimbate) dintre participanții la *chat* furnizează gradul de centralitate (*degree centrality*). Această informație este folositoare în cadrul analizelor rețelei şi mai ales a relațiilor sociale din cadrul ei. Însă aceasta este doar partea cantitativă, ea fiind sprijinită de o analiză calitativă, şi anume ce fel de replici circulă între participanți, ce exprimă acestea fiind mai degrabă tratate mai mult teoretic deocamdată.

D. Centralitate

Conform teoriei grafurilor există 5 tipuri de centralități, de:

- apropiere („closeness")
- graf
- trecere („betweenness")
- stress
- valori proprii („eigenvector")

Apropierea

În domeniul topologiei, apropierea este unul din conceptele de bază. Intuitiv, două seturi sunt apropiate dacă sunt vecine unul cu celălalt. Conceptul poate fi definit în mod natural într-un spațiu metric unde noțiunea distanței dintre două elemente este definită, dar poate fi generalizată la spații topologice unde nu există o măsură concretă a distanței.

În teoria grafurilor, *apropierea* este o măsură a centralității unui nod într-un graf. Ea este invers proporțională față de distanța minimă dintre nodul curent şi toate celelalte noduri. Astfel nodurile centrale dintr-un graf au o valoare mai mare a apropierii [1]:

$$c_C(v) = \frac{1}{\sum_{t \in V \setminus v} d_G(v,t)}$$

unde $d_G(v,t)$ reprezintă distanța minimă dintre nodurile v şi t.

Centralitatea grafului

Într-o rețea este importantă distanța maximă între nodul curent şi toate celelalte noduri, pentru a determina apropierea relativă între participanții la *chat*. Atât pentru calcularea *centralității grafului* cât şi pentru apropiere se poate folosi *algoritmul Floyd-Warshall* [1,2]:

$$c_G(v) = \frac{1}{\max_{t \in V \setminus v} d_G(v,t)}$$

Centralitatea de trecere

O altă măsură a centralității într-un graf este *centralitatea de trecere,* conform căreia au un grad mai mare nodurile care se află pe mai multe drumuri minime din graf dintre oricare alte două noduri [1]:

$$c_B(v) = \sum_{s \neq v \neq t \in V} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

unde σ_{st} reprezintă numărul de drumuri minime între nodurile s şi t; $\sigma_{st}(v)$ - numărul de drumuri minime între s şi t care trec prin nodul v.

Factorul de stress

În cazul factorului de *stress*, centralitatea este corelată cu suma tuturor drumurilor minime din graf care trec prin nodul selectat [1]:

$$c_S(v) = \sum_{s \neq v \neq t \in V} \sigma_{st}(v)$$

Valori proprii

Centralitatea obținută prin calculul valorilor proprii este o altă măsură de determinare a importanței unui participant din rețea. Acest factor atașează note relative tuturor nodurilor din rețea pe baza principiului că o legătură cu un nod de rang înalt este mai importantă decât un număr de legături cu noduri de rang mai mic.

E. Rang

Folosind algoritmul Page Rank [7], care stă la baza sistemului Google, este determinat rangul unui utilizator, având la bază matricea cu numărul de intervenții schimbate între participanți, respectiv pe notele replicilor interschimbate. Cu cât un utilizator este mai căutat, înseamnă că informațiile provenite de la el sunt mai valoroase pentru ceilalți participanți, deci rangul lui va fi crescut pe măsură ce ii sunt adresate mai multe replici de la persone, preferabil, cât mai importante. În această evaluare este foarte importantă ponderea personei care comunică cu utilizatorul (dacă o persoană cu un rang mare comunică cu utilizatorul, atunci probabilitatea ca informația exprimată să fie importantă este ridicată). Pentru calcularea rangului se foloseşte o metodă iterativă pentru evaluarea sistemului format din ecuațiile:

$$UR(A) = (1-d) + d\left(\frac{UR(t_1)}{c(t_1)} + \ldots + \frac{UR(t_n)}{c(t_n)}\right),$$

unde *UR*= user rank – rangul unui participant;

C_i = numărul de replici interschimbate de participantul t_i cu participantul A;

d = o constantă (0.85) folosită pt o convergența cât mai rapidă a sistemului.

F. Nota intervenției

Se face presupunerea că fiecare conversație are un anumit tipar pe care il urmează: introducerea subiectului conversației, partea mediană, unde se găsesc firele de discuție (pot exista unul sau mai multe astfel de fire în *chat*) şi sunt zone efective de colaborare intensă, şi apoi finalul, în care se regăsesc concluziile discuției.

Revenind la dezbaterea dintre calitate, cantitate şi eficiență, calculul notei intervenției se face nu numai pe

baza cuvintelor folosite, ci şi pe baza lungimii interventiei. Prin cuvinte, se înţeleg doar acei termeni care sunt relevanţi discursului, şi nu cuvintele care nu aduc conţinut, cum ar fi articolele sau conjuncţiile (”stopwords”), care sunt eliminate din analiză.

În evaluarea unei replici, pe lângă lungimea propriu-zisă se iau în considerare şi următorii factori:

- numărul cuvintelor cheie, care rămân din replică după corectarea erorilor (spellcheck), extragerea rădăcinii (stemming) şi eliminarea ”stopwords” – *lungimeMC*;
- nivelul la care se află intervenţia participantului în înşiruirea replicilor.

Astfel s-a obţinut formula de calcul:

$$nota_{empirica} = \frac{\frac{lungime}{6} + \frac{lungimeMC * 5}{6}}{1 + \frac{1}{nivel}}.$$

Pornind de la această notă empirică se determină nota finală a replicii:

$$nota_{finala} = nota_{replica\ precedenta} + coeficient * nota_{empirica},$$

unde coeficientul este determinat în funcţie de tipul replicii curente şi cel al replicii de care aceasta e legată.

Un rol foarte important în alegerea coeficientului o are identificarea tipurilor de acte de vorbire din fiecare replică [9]. În această direcţie se urmăreşte identificarea verbelor din intervenţie, prezenţa anumitor semne de punctuaţie (de întrebare) şi a anumitor cuvinte cheie.

Se obţine astfel o listă de atribute pentru o replică şi, ţinând cont de atributele replicii precedente (de care replica curentă este corelată) se determină coeficientul. În implementarea actuală există o matrice din care se selectează aceste valori, atributele luate în considerare fiind: negaţie, confirmare, întrebare şi afirmaţie.

În evaluarea unei replici, în cazul analizei de tip semantic a reţelei sociale se va realiza o filtrare a replicilor din punctul de vedere al înrudirii termenilor folosiţi cu o listă de cuvinte introduse de utilizator sau cu o listă de cuvinte cheie ale *chat*-ului determinată statistic. Pentru a determina aceste cuvinte cheie se creează domenii de cuvinte inrudite, folosind relaţii de tip sinonimie din WordNet (http://wordnet.princeton.edu) sau LSA (”Latent Semantic Analysis” - Analiza Semantica Latenta, vezi http://lsa.colorado.edu), iar apoi, pe baza frecvenţei cuvintelor, a poziţiei în replică şi a importanţei totale a replicii se obţine lista de candidaţi ordonaţi după aceste criterii.

Din noţiunile prezentate mai sus, pentru o înşiruire de replici aparţinând aceluiaşi fir de discuţie din *chat* se obţine o evoluţie ca cea reprezentată în figura 3.

Prezenţa valorilor negative, se datorează unei note empirice mari a replicii curente şi a unui coeficient negativ, obţinut din calculul atributelor replicii respective şi celei precedente.

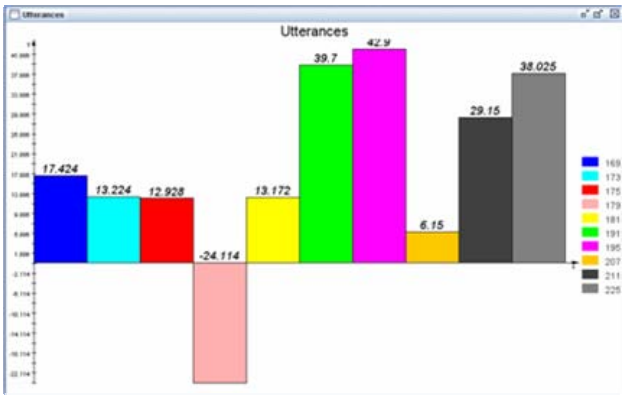


Figura 3 Evoluţia replicilor dintr-o înşiruire

Asemănător se determină şi evoluţia replicilor din întregul *chat*: în care se iau în considerare toate firele de discuţie, fiind posibilă obţinerea unor valori negative în cazul unor anumite succesiuni de atribute ale replicilor.

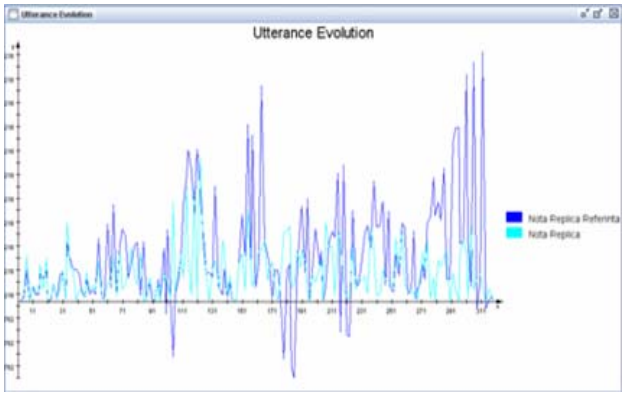


Figura 4 Evolutia chatului

Folosind aceeaşi idee se obţine şi evoluţia totală a fiecărui participant în funcţie de notele replicilor sale. În acest caz se iau în considerare doar notele empirice ale replicilor şi nu cele finale pentru a se obţine o repartiţie cât mai corectă a implicării fiecărui participant.

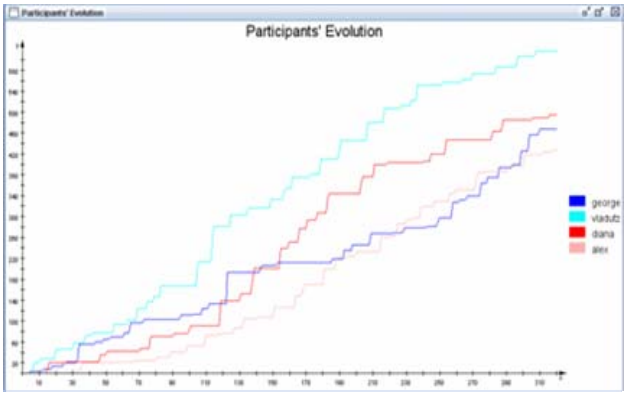


Figura 5 Evoluţia participanţilor

Pentru primii doi factori (numărul de caractere şi numărul de caractere raportat la numărul de intervenţii) se ţine cont

doar de matricea construită pe baza numărului de replici interschimbate.

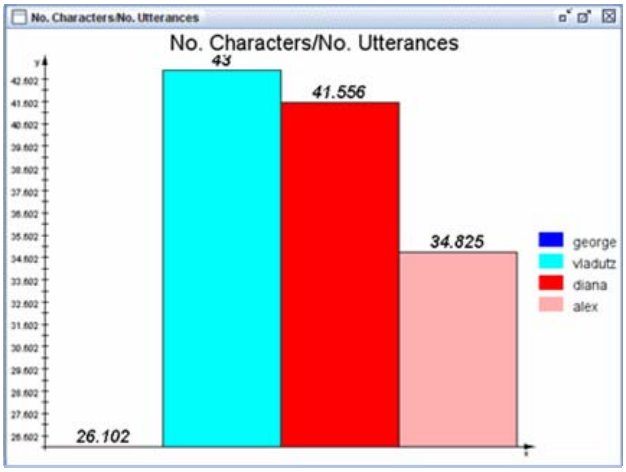
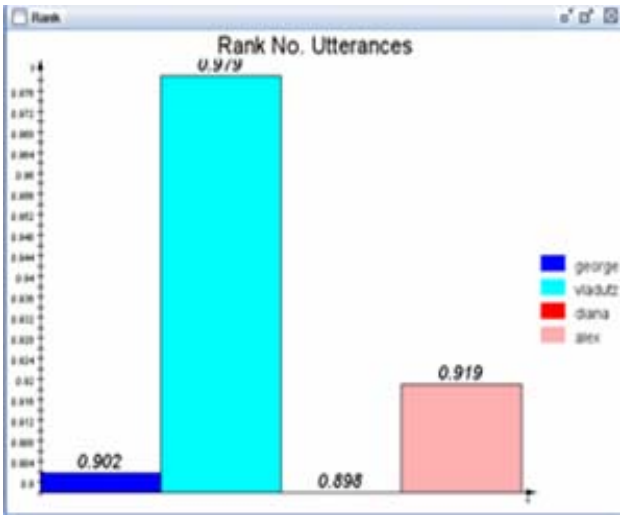


Figura 4 Grafice pentru primii doi factori



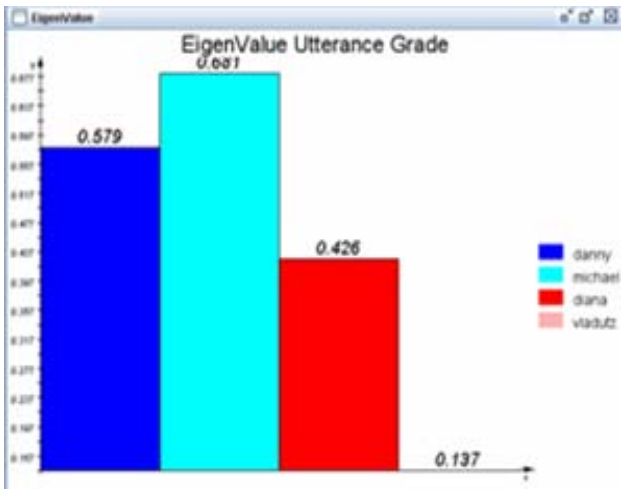


Figura 5 Grafice în funcţie de factorul ales şi indicatorul la care sunt raportate: numărul/nota replicilor

Pentru fiecare din ceilalţi factori se generează două grafice: unul raportat la numărul de replici interschimbate şi al doilea în funcţie de notele replicilor.

G. Analiza Semantică Latentă (LSA)

LSA (http://lsa.colorado.edu) este o tehnică utilizată în procesarea limbajului natural, în particular în semantica vectorială a analizei relaţiilor dintre un set de documente şi termenii pe care îi conţin, producând o serie de concepte în relaţie cu documentele şi termenii conţinuţi de fiecare.

Această tehnică foloseşte o matrice rară care conţine pe coloane documentele care pot fi căutate, iar pe linii termenii (de obicei rădăcina cuvintelor cheie după care se face căutarea) conţinuţi în aceste documente. Elementele matricei sunt de obicei ponderea *tf-idf (term frequency – inverse document frequency)*, adică este proporţional cu frecvenţa termenului în colecţia de documente şi este invers proporţional cu numărul de documente în care apare (astfel termenii rari având ponderi mai mari, iar cei care apar în foarte multe documente ponderi mai mici [4]. *LSA* transformă, prin metode algebrice, această matrice în relaţii între termeni şi documente precum şi în grupuri de cuvinte înrudite (aşa numitele ”spaţii semantice”).

Un alt lucru important în evaluarea *chat*-ului este determinarea *topicelor* pentru care, pe lângă folosirea WordNet-ului ca ontologie se apelează şi la relaţiile dintre cuvinte obţinute în urma aplicării algoritmului *LSA*.

IMPLEMENTARE

Aplicaţia a fost realizată în JAVA 1.6, iar elementele grafice sunt de tip SWING.

Primele probleme care au apărut la prelucrearea unui *chat* au fost cele legate de ortografie: faptul că discuţiile sunt purtate *on-line* (perifericele folosite sunt tastaturile) face ca probabilitatea apariţiei erorilor să fie ridicată. Un alt factor este tendinţa de a folosi abrevieri pentru a scurta timpul necesar tastării, sau folosirea de *emoticoane* pentru definirea anumitor stări sufleteşti. Astfel, corectarea cuvintelor scrise greşit (*spellchecking*) este unul dintre principalele obiective (nu numai în proiectul curent, dar şi în cazul proiectelor similare, care au ca obiectiv analiza limbajului natural).

Pentru *spellchecking* se foloseşte biblioteca Jazzy (vezi http://jazzy.sourceforge.net/ şi http://www.ibm.com/developerworks/java/library/j-jazzy/), care este foarte asemănătoare cu Aspell, îmbinând algoritmi de *potrivire fonetică* cu algoritmi de tip comparare de secvenţe. S-au realizat şi următoarele îmbunătăţiri:

- se sparge cuvântul iniţial în două subcuvinte, se aplică *spellchecking* pe fiecare, şi apoi, folosind distanţa Levenshtein, se determină dacă nu cumva cuvântul iniţial provine din unirea a două cuvinte (erorile obtinute prin alipirea a două cuvinte/omiterea spaţiului dintre ele sunt frecvente intr-un *chat*, iar Jazzy, în versiunea iniţială nu determină astfel de greşeli de scriere);
- se construieste un graf folosind un dicţionar în care se memorează frecvenţa de apariţie a fiecărui cuvânt: în cazul unei greşeli de scriere, pentru determinarea celei mai bune sugestii din lista oferită de Jazzy (în care se iau în considerare atât asemănările fonetice cât şi distanţa Levenshtein) se ţine cont şi de frecvenţa apariţiilor fiecărei sugestii.

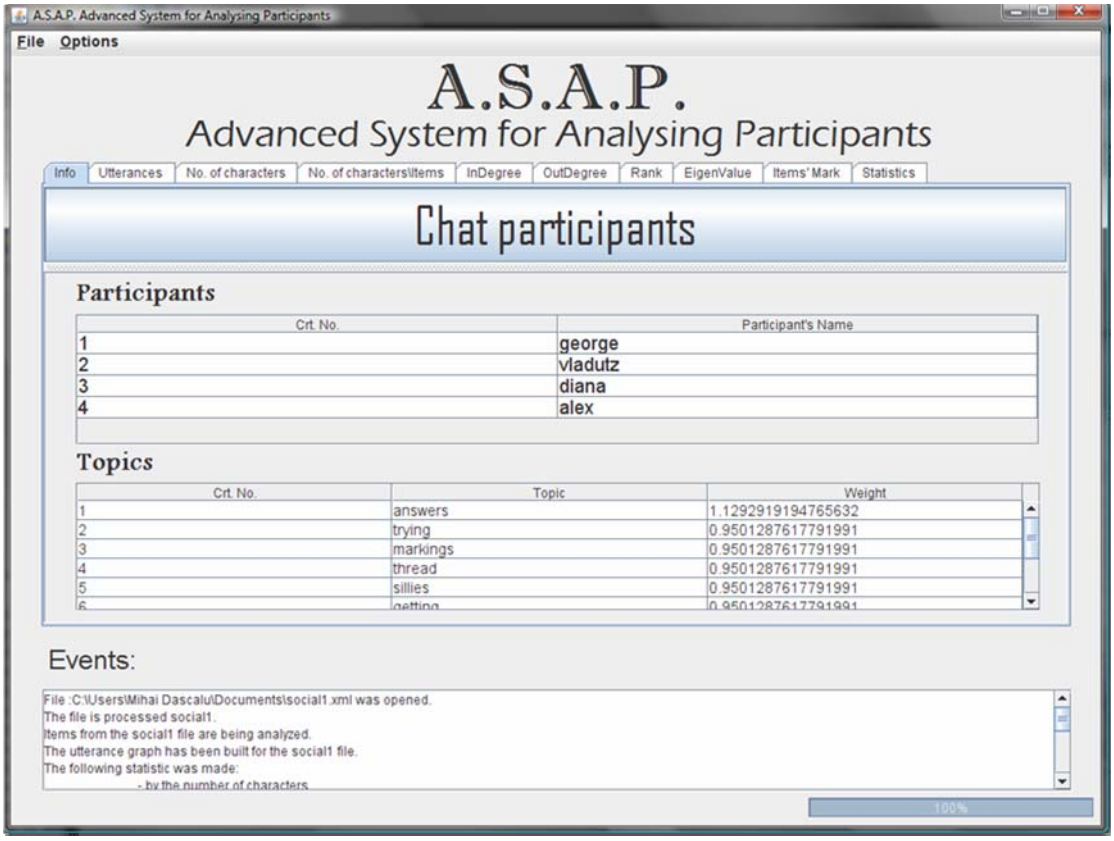


Figura 6 Interfaţa A.S.A.P.

După cum s-a precizat anterior, o simplificare uzuală în analiza textelor este analiza doar a cuvintelor importante, care conţin înţeles, care dau fondul formei. În consecinţă, toate cuvintele de legătură (conjuncţii, prepoziţii, articole, interjecţii ş.a.m.d.) – aşa numitele *stopwords* - sunt eliminate din start.

Etapa următoare presupune aducerea cuvintelor rămase la forma lor de rădăcină, fără flexiuni, astfel încât căutarea lor într-un lexicon să fie mai uşoară (*stemming*). Deşi acest pas este opţional în căutare – *Wordnet*-ul acceptă căutarea cuvintelor în orice formă – este absolut necesară în cazul identificării spaţiilor semantice cu LSA.

Stemmer-ul folosit este *Snowball* (http://snowball.tartarus.org/), care oferă cele mai bune rezultate în comparaţie cu *Porter* (http://tartarus.org/~martin/PorterStemmer/) sau *Lovin Stemmer*, fiind bazat tot pe un algoritm, şi nu pe un dicţionar – performanţe mult mai bune în condiţiile în care se foloseşte iniţial un spellchecker pentru corectarea cuvintelor.

Din punctul de vedere al ontologiilor s-a folosit *Wordnet* (http://wordnet.princeton.edu), acesta fiind un lexicon semantic al limbii engleze, care include cuvintele în grupuri de sinonime denumite *synsets* şi furnizează diferite relaţii semantice între aceste seturi. Utilizarea Wordnet-ului permite descoperirea actelor de vorbire prin identificarea verbelor şi corelarea replicilor într-un context dat.

Pentru calculul notei unui fir de conversaţie, fiecare replică poate duce la creşterea sau scăderea valorii acesteia. Astfel, în funcţie de tipul intervenţiei sau tipul actului de vorbire, nota are o pondere negativă sau pozitivă. În acest moment, implementarea acestei proprietăţi este oarecum superficială, urmând ca în urma analizei corpusurilor colectate, să se creeze un „tabel de intervenţii" – acesta conţine ponderi pentru fiecare tip de intervenţie majoră (negaţie, întrebare, afirmaţie s.a.m.d.).

Generarea grafurilor foloseşte JopenChart Drawer (http://jopenchart.sourceforge.net/), care generează imagini JPEG ale grafurilor.

La implementarea calculului centralităţii cu ajutorul vectorilor şi valorilor proprii a fost introdusă o bibliotecă pentru calcul algebric, JAMA (http://math.nist.gov/javanumerics/jama), cu ajutorul căreia s-a realizat descompunerea matricilor.

Coeficienţii calculaţi ajută la creearea unui top descrescător per coeficient şi unui top descrescător după nota finală a fiecărui participant calculată după o formulă în care coeficienţii factorilor prezentaţi anterior au diferite ponderi.

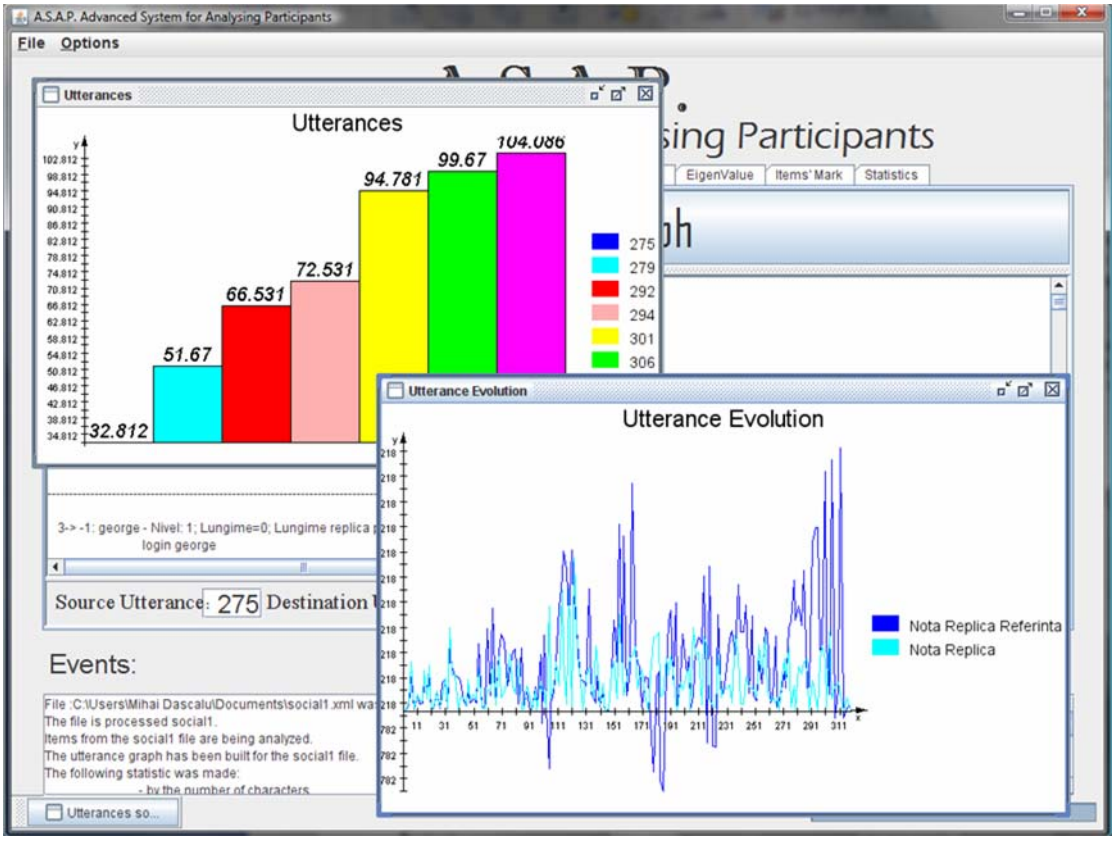


Figura 7 Vizualizarea ferestrelor din cadrul aplicației A.S.A.P

ADNOTAREA CORPUSULUI PENTRU ANALIZĂ

Pentru a valida si a regla parametrii evaluării participanților la un *chat*, a fost dezvoltat un program de asistare a adnotării *chat*-urilor efectuate. Acesta a fost folosit în cadrul câtorva cursuri. Astfel a fost posibilă obținerea unui corpus adnotat pe baza căruia să se compare rezultatele obținute folosind A.S.A.P., sistemul nostru, şi cele obținute în urma evaluării *chat*-ului de o persoană umană.

Scopul construirii acestui corpus, a unui "*standard de aur*", este acela de a vedea cât de aproape este aproximarea matematică făcută de program de o aproximare făcută de un om. Practic, aceasta etapă, care pune algoritmii la încercare, este una dintre cele mai surprinzătoare şi mai incitante dintre toate etapele parcurse.

La momentul actual, corpusul adnotat a fost construit şi aşteaptă analiza care va aduce cu ea concluziile privind performanța şi eficiența metodelor descrise anterior. Aplicatia este realizata tot în Java 1.6 si de asemenea integrează elemente grafice de tip SWING.

Aplicația permite introducerea următoarelor informații cu privire la *chat*-ul analizat, care apoi vor fi salvate sub forma unui XML:

- crearea de adnotări pe baza unei replici: comentarii, note instant pentru o replică oarecare;
- adăugarea de note pentru fiecare participant pentru o înşiruire de 20 de replici succesive, reflectând activitatea sa din respectiva zonă;
- note finale pentru participanții la *chat* şi pentru relevanța *chat*-ului: colaborarea dintre participanți şi rămânerea "on topic";
- adăugarea de legături implicite incluzând tipurile referințelor şi şabloanele identificate;
- în funcție de evoluția *chat*-ului, cuvintele cheie/relevante de pe parcurs (topic-urile);
- specificarea zonelor de colaborarea intensă în care informațiile oferite sunt relevante raportat la discuția purtată

Figura 8 Interfata Adnotator

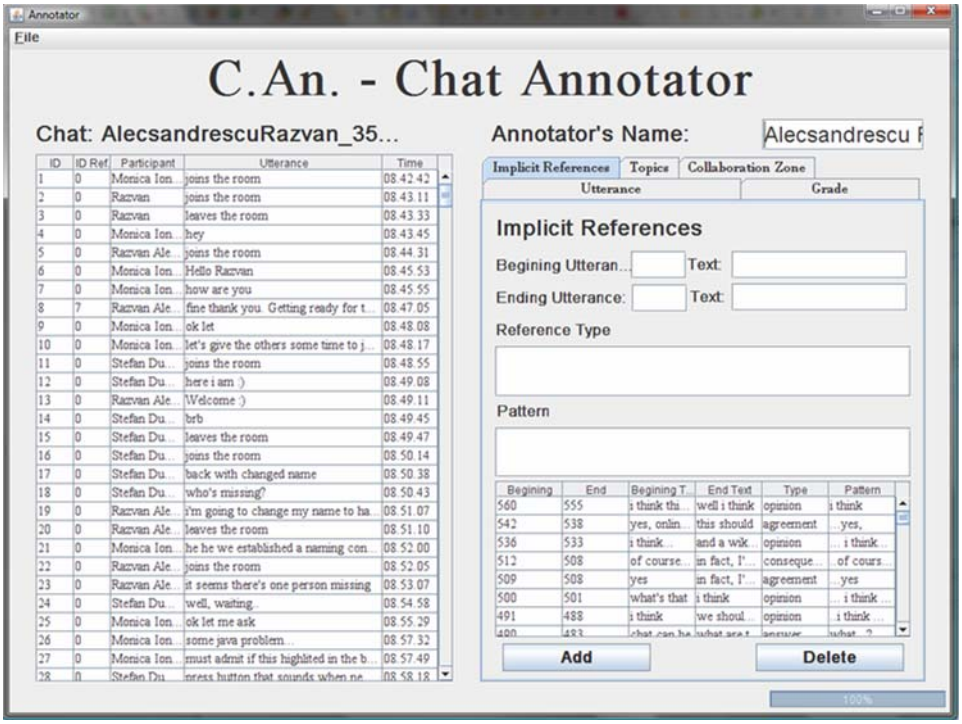


Figura 9 Deschiderea unui fisier de adnotare pentru completarea informațiilor deja existente

CONCLUZII ŞI VIITOARE ÎMBUNĂTĂŢIRI

ASAP este o interfaţă foarte utilă în examinarea interacţiunilor om-calculator folosind sisteme de chat, nu numai în aplicaţiile educaţionale, ci şi în, de exemplu, proiectare sau rezolvare de probleme colaborative.

Primele rezultate ne încurajează să conchidem că se poate determina automat competenţa unei persoane care participă la un chat. În viitor preconizăm să facem următoarele înbunătăţiri:

- determinarea replicilor/referinţelor implicite folosind Wordnet-ul şi LSA-ul prin găsirea asocierilor între cuvintele din acelaşi câmp semantic şi din replici diferite;
- implementarea modelului probabilistic de analiză semantică latentă (PLSA, vezi http://www.cs.brown.edu/people/th/papers/Hofmann-UAI99.pdf) şi compararea rezultatelor cu cele obţinute folosind LSA-ul;
- prelucrarea distribuită a datelor, astfel oferind posibilitatea folosirii programului pe sisteme distribuite/grid care permit în acest caz optimizări notabile;
- determinarea listelor de cuvinte aparţinând aceluiaşi domeniu, totodată păstrând nivelul de generalitate şi câmpul lexical din care face parte;
- considerarea influenţei notei replicii / poziţia unui cuvânt cheie în intervenţie pentru îmbunătăţirea notei replicilor;
- contopirea domeniilor obţinute folosind *LSA*-ul în funcţie de domeniu/câmp semantic în vederea obţinerii zonelor cu cel mai mare interes/pondere din *chat*; se poate folosi şi o abordare statistică în care să se considere şi contopirea domeniilor pe baza existenţei unui sinonim comun sau o relaţie ierarhică folosind *WordNet*-ul;
- folosirea unui corpus mai mare pentru antrenarea LSA;
- după crearea unui set mai amplu de *chat*-uri evaluate, aplicarea unei indexări inverse pentru determinarea celui mai competent participant din mai multe *chat*-uri pe baza unor anumite criterii, eventual folosirea modelului *Map Reduce* sau a unor elemente specifice;
- îmbunătăţirea determinării tipului replicii şi identificării actelor de vorbire pentru corelarea intervenţiilor într-o înşiruire.

PRECIZĂRI

Această lucrarea a fost parţial finanţată din proiectul European FP7 STREP LTfLL şi din proiectul naţional CNCSIS K-Teams.

REFERINŢE

1. Brandes, U. (2001), ”A Faster Algorithm for Betweenness Centrality”, *Journal of Mathematical Sociology* 25(2):163-177.
2. Cormen, T., Leiserson, C., Rivest, R., Stein, C., (2001), *Introduction to Algorithms*, MIT Press.
3. Holmer, T., Kienle, A. & Wessner, M. (2006) Explicit Referencing in Learning Chats: Needs and Acceptance, in Nejdl, W., Tochtermann, K., (eds.), *Innovative Approaches for Learning and Knowledge Sharing, First European Conference on Technology Enhanced Learning, EC-TEL 2006*, *Lecture Notes in Computer Science*, 4227, Springer, pp. 170-184.
4. Manning, C., Schutze, H. (1999*) Foundations of Statistical Natural Language Processing*, MIT Press: Cambridge Mass.
5. Newman, M. E. J., The mathematics of networks, http://www-personal.umich.edu/~mejn/papers/palgrave.pdf
6. Ciprian Onofreiciuc, Alexandru Roşiu, Alexandru Gartner, Ştefan Trăuşan-Matu, „Polyphony, a Knowledge-based Chat System Supporting Collaborative Work”, în C. Badica, M. Paprzycki (Eds.), „Advances in Intelligent and Distributed Computing”, Proceedings of IDC, Studies in Computational Intelligence Vol. 78, Springer, 2007, pp. 155-164.
7. Page, L., Brin, S., Motwani, R., and Winograd, T., *The pagerank citation ranking: Bringing order to the web*. Technical report, Stanford Digital Library Technologies Project, 1998
8. Stahl, G. (2006) *Group Cognition: Computer Support for Building Collaborative Knowledge,* MIT Press.
9. Trausan-Matu, S., Chiru, C., Bogdan, R., (2004), Identificarea actelor de vorbire în dialogurile purtate pe chat, în Stefan Trăusan-Matu, Costin Pribeanu (Eds.), *Interactiune Om-Calculator 2004*, Editura Printech, Bucureşti, pp. 206-214.