

Sistem de recomandare de imagini pe baza aspectelor sociale, semantice și vizuale

Maria Roditis¹, Andreea Tabacariu¹, Ștefan Trăușan-Matu^{1,2}

¹ Universitatea „Politehnica” București
Splaiul Independenței nr. 313, sector 6, București
E-mail: roditis.maria@gmail.com, andreea.tabacariu@gmail.com

² Institutul de Cercetări în Inteligență Artificială,
Calea 13 Septembrie nr. 13, București
E-mail: stefan.trausan@cs.pub.ro

Rezumat. Articolul are ca scop prezentarea sistemelor de recomandare prin prisma celor mai importante aspecte și tipuri de algoritmi folosiți în diverse abordări și implementări, precum și a problemelor ce apar și trebuie preîntâmpinate. Aplicația prezentată în acest articol utilizează o abordare care combină multiple modalități de recomandare de imagini. Modelul propus aduce rezultate calitative pe baza aspectelor: sociale - prin aplicarea conceptului de filtrare colaborativă, vizuale - prin calculul de similarități între trăsăturile vizuale ale imaginilor, semantice - prin utilizarea metricilor de similaritate între termeni bazate pe WordNet (<http://wordnet.princeton.edu>), pe care le integrează cu utilizarea unui clasificator SVM pentru recunoașterea tipurilor de scenă pe baza aspectelor vizuale. Prezentăm în detaliu metodele științifice și soluția arhitecturală care stau la baza realizării sistemului propus.

Cuvinte cheie: recomandare de imagini, clasificare, trăsături vizuale, semantice, sociale

1 Introducere

Sistemele de regăsire de imagini, sistemele de recomandare, precum și detecția de concepte vizuale și adnotarea automată a imaginilor reprezintă direcții de cercetare care în ultimii ani au primit și continuă să primească multă atenție la nivel academic și nu numai, ele extinzând remarcabil capacitățile mașinilor de căutare de tip google. De exemplu, ACM organizează în 2011 a cincea conferință pe tema sistemelor de recomandare (<http://recsys.acm.org>). De asemenea, sistemele de recomandare au fost declarate ca fiind una dintre primele cinci teme de interes deosebit în domeniul *web*-ului, printre altele precum *web*-ul semantic, realitatea îmbunătățită, *web*-ul mobil și rețelele sociale (MacManus, 2009).

Sistemele de recomandare sunt aplicații software concepute pentru a filtra informația și a oferi sugestii utilizatorilor despre obiecte ce ar putea fi de interes pentru ei, pe baza unor anumite criterii. Multe din aplicațiile *online* care folosesc sau doresc să includă o astfel de soluție au milioane de înregistrări stocate în diverse medii, conținând informații despre articolele cu care aplicația lucrează, despre utilizatorii săi și despre activitățile realizate de aceștia. Sistemul rezultat trebuie să ofere scalabilitate și procesare în timp real a cererilor de recomandare, pentru a facilita utilizatorului final o experiență agreabilă.

Competiții internaționale precum *ImageCLEF* încearcă să încurajeze cercetarea în domeniul adnotării automate a imaginilor, pe baza detecției conceptelor vizuale, prin punerea la dispoziție a unui set de date comun și unor metodologii de evaluare a performanțelor. (<http://www.imageclef.org>) Această competiție face posibilă evaluarea comparativă a mai multor algoritmi și stimulează dezvoltarea și îmbunătățirea metodelor de adnotare automată a imaginilor. În conjuncție cu aceste competiții deschise, activitatea de cercetare în aceste domenii se concretizează printr-un număr mare de articole, noi abordări și noi metode de evaluare. Cu toate acestea, progresul în domeniul regăsirii și recomandării de imagini este departe de a fi ajuns la un nivel final.

2 Descrierea aspectelor teoretice asociate sistemului propus

2.1 Sisteme de recomandare

Sistemele de recomandare sunt sisteme care oferă facilități de predicție a preferințelor utilizatorilor într-un spațiu de articole dat, pentru a-i ajuta în procesul de selecție a articolelor (Resnick și Varian, 1997). După algoritmi utilizați, aceste sisteme se împart în două mari categorii.

Primul tip de sisteme de recomandare utilizează recomandarea bazată pe conținutul articolului. Acești algoritmi se axează pe conținutul efectiv al articolului și oferă ca recomandări articole similare unui articol inițial pe baza unor metrice specifice (de pildă: tag-uri, tema articolului, etc). Această abordare este cuprinsă în aplicația noastră prin recomandările făcute pe bază de etichete (*tag-uri*) și pe baza trăsăturilor din imagine. Punctul slab al acestei abordări este că nu ia în considerare și partea socială, ce constă în modele de similaritate între profilurile de preferințe ale utilizatorilor.

Cea de-a doua modalitate de abordare, filtrarea colaborativă, utilizează entitățile din aplicație (utilizatori, articole, preferințe, acțiuni șamd) și recomandă articole pe baza preferințelor utilizatorilor. Preferințele utilizatorilor pot fi obținute prin exprimarea lor explicită, prin vot, sau implicită, prin studiul comportamentului utilizatorilor. Metodele de filtrare colaborativă se pot clasifica la rândul lor în două categorii: (1) metode bazate pe utilizator: se folosesc preferințele utilizatorilor similari pentru un articol dat pentru a descoperi recomandări de articole, similaritatea între utilizatori determinându-se pe baza preferințelor exprimate în sistem (utilizatorii similari având, informal spus, „aceleași gusturi”); precum și (2) metode bazate pe articol: se găsesc articolele similare unui articol dat prin cuantificarea anumitor aspecte sociale, de pildă faptul că mulți dintre utilizatorii care au preferat articolul dat, au votat pozitiv și un alt articol.

2.2.1 Caracteristici ale sistemelor de recomandare

Sistemele de recomandare sunt caracterizate prin multiple criterii, iar unele dintre cele mai importante dintre acestea sunt detaliate în continuare (Riedl, Beupre, Sanders, 2009).

Transparența recomandării reprezintă o calitate prin care sistemul oferă o motivație pentru recomandarea unui anumit articol (de ex: „*Pentru că v-a plăcut filmul Nașul, vă recomandăm și Lista lui Schindler*”), iar astfel poate crește încrederea unui utilizator în calitatea algoritmului de recomandare utilizat.

Un alt aspect este legat de *explorare versus exploatare*. Presupunem situația în care există două articole care pot fi expuse (recomandate) unui grup de utilizatori și se poate înregistra activitatea pentru un articol inițial expus. Problema la care trebuie să se răspundă este *când și dacă* ar trebui să fie expus și al doilea articol, așa încât impactul dorit asupra utilizatorului să fie maxim (ex: număr de vizualizări, volum de vânzări etc), și este cunoscută în inteligența artificială ca și problema „*multi-armed bandit*” (mai multe detalii despre ea în Vermorel și Mohri, 2004). Problema constă în faptul că un jucător are la dispoziție un joc mecanic de tip *slotmachine* cu mai multe variante (manete), iar el dorește o strategie de tragere a manetelor astfel încât să-și maximizeze profitul total.

Ghidarea navigării reprezintă problema oferirii unei interfețe adecvate utilizatorului, atunci când există un conținut foarte vast în cadrul aplicației,

pe baza unor tehnici de interacțiune om-calculator, a unor algoritmi inteligenți, a prezicerii intenției sau dispoziției utilizatorului șamd.

Prin *valorificarea corectă a momentului de timp*, un sistem de recomandare ține cont de schimbările pe care trecerea timpului le aduce asupra relevanței unui articol, precum: creșterea interesului pentru articole ce privesc evenimente recente, posibilitatea ca sistemul de recomandare să asimileze rapid informație nouă etc.

Scalabilitatea sistemului este deosebit de importantă în contextul gestiunii de volume mari de date și în continuă creștere. În plus, algoritmi de recomandare sunt în general iterativi, bazați pe procesarea extensivă a datelor. Sunt de dorit timpi de răspuns mici pentru utilizatorul final, iar în acest sens sunt folosite procesări *offline*, și în mică parte procesări *online*, la momentul cererii formulate de utilizator.

Diversitatea constituie, potrivit lui (Linden, 2009), un aspect de dorit în unele cazuri de recomandare, atunci când recomandarea de elemente aproape duplicate minimizează impactul sau utilizabilitatea sistemului pentru utilizator.

2.2.2 Probleme clasice ale sistemelor de recomandare

Problema „*cold start*” apare în cazul utilizatorilor noi ai sistemului, atunci când nu există informație explicită despre preferințele acestora pentru a produce recomandări. Printre soluțiile posibile se numără: completarea de chestionare legate de preferințe – nerecomandat întrucât un sistem de recomandare inteligent ar trebui să ceară explicit cât mai puține informații din partea utilizatorului – sau utilizarea de agenți de învățare care să descopere profilul utilizatorului pe baza acțiunilor realizate de acesta. Problema „*first rater*” este problema echivalentă în cazul articolelor nou intrate în sistem, pentru care nu există încă preferințe exprimate. O soluție viabilă în acest caz este realizarea unui mecanism hibrid de recomandare, care poate face recomandări bazate și pe conținutul efectiv al unui articol pentru a descoperi articole similare.

În ce privește utilizatorul însuși, apare *problema manipulării (influențării)* sistemului de recomandare de către un grup mic de utilizatori, de exemplu prin comentarii negative la adresa unui articol oferit de competitori. În acest sens, trebuie ca sistemul să discearnă și să elimine aceste tentative de manipulare. Trebuie evaluat de asemenea și *impactul recomandării*, constând în costul furnizării unei recomandări proaste sau

omiterii unei recomandări bune, atunci când alegerea unui articol are impact important asupra utilizatorului. *Problema confidențialității* apare în condițiile în care trebuie ca un utilizator să poată evalua recomandările primite, dar nu se dorește ca preferințele utilizatorilor care au condus la elaborarea recomandării să fie publice sau să poată fi determinate prin manipularea datelor de intrare ale sistemului.

2.2.3 Evaluarea sistemelor de recomandare

Evaluarea performanțelor algoritmilor de recomandare se poate realiza prin utilizarea unui spectru larg de metode și metrici (Shani și Gunawardana, 2011).

O măsură a erorii foarte folosită este *root mean square error (RMSE)*, care se calculează pentru preferințele utilizatorilor prin următoarea formulă:

$$RMSE(p) = \sqrt{\frac{\sum_i (p_{est,i} - p_{real,i})^2}{n}}$$

Altă măsurătoare a erorii constă în realizarea mediei diferenței absolute între valoarea reală a preferinței și valoarea estimată de sistemul de recomandare.

Evaluarea sistemelor de recomandare se mai poate realiza și prin împărțirea datelor într-un set de antrenare și un set de testare în maniera în care se face și evaluarea clasificatorilor, deși este o metoda mai puțin frecventă. Sistemului îi este prezentat setul de antrenare și apoi trebuie să calculeze preferințele pentru instanțele din setul de testare. Aceste preferințe sunt comparate cu cele reale și se obține astfel un scor de eroare pentru acest algoritm.

Însă nu corectitudinea în întreg setul contează (Linden, 2009), măsurată prin RMSE, de exemplu, ci uneori este de dorit minimizarea erorii doar în recomandările care intră în „Top N”. Cu alte cuvinte, sistemul de recomandare dorește să aibă precizie maximă în determinarea celor mai probabile preferințe și nu a articolelor mai puțin probabil să fie preferate.

2.2.4 Aplicații și lucrări științifice similare

Amazon

Amazon.com folosește algoritmi de recomandare pentru a personaliza magazinul *online* pentru fiecare client. Potrivit prezentării oferite, algoritmul, cunoscut sub numele *Item-to-Item Collaborative Filtering* (Linden, Smith și York, 2003), oferă recomandări în timp real, este scalabil la un set de date mare și obține recomandări bune. Descrierea detaliată a soluției nu este disponibilă pe internet, dar din diverse documente ce prezintă aplicația se pot extrage câteva idei, precum și algoritmul la nivel înalt. În loc să creeze profilul utilizatorilor și să facă seturi de utilizatori similari, sistemul de recomandare de la Amazon face legătura între fiecare obiect cumpărat de utilizator și produse notate de utilizatori similari și apoi combină acele produse similare într-o listă de recomandări. Pentru a determina cea mai bună potrivire pentru un obiect dat, algoritmul construiește o tabela de obiecte similare, găsind produse pe care consumatorii tind să le cumpere împreună. Se poate construi o matrice produs-produs iterând toate perechile și calculând o metrică de similaritatea pentru fiecare pereche. Cu toate acestea, multe perechi de produse nu au clienți comuni, și de aceea abordarea este ineficientă din punctual de vedere a timpului de procesare și al memoriei folosite. Următorul algoritm iterativ oferă o abordare mai bună, calculând similaritatea între un singur produs și toate produsele care au o legătură cu produsul de referință.

Fie $Produse = \{ I_i \mid I_i \text{ un produs} \}$, $Cienti = \{ C_i \mid C_i \text{ un client} \}$.

Fie funcțiile auxiliare:

$CumparatoriPentru : Produse \rightarrow P_{Cienti}$, a.i.
 $CumparatoriPentru(I_i) = \{ C_{i1} \dots C_{ij} \dots C_{in} \mid C_{ij}, \text{ cumparator al produsului } I_i \}$,
 $\forall I_i \in Produse \text{ și } C_{ij} \in Cienti$.

$CumparateDe : Cienti \rightarrow P_{Produse}$ a.i. $CumparateDe(C_i) = \{ I_{j1} \dots I_{jk} \dots I_{jm} \mid I_{jk} \text{ a fost cumparat de clientul } C_i \}$, $\forall C_i \in Cienti \text{ și } I_{jk} \in Produse$.

$CalculSimilaritate : Produse \times Produse \rightarrow [0,1]$ a.i. $CalculSimilaritate(I_i, I_k)$ este rezultatul aplicării unei metrici de similaritate M între două produse I_i și I_k .

```

Funcția CalculeazaSimilaritati (Catalog) {
    Fie Asocieri :=  $\emptyset$ , Similaritati :=  $\emptyset$ 
    Pentru fiecare produs  $I_i$  în Produse {
        Pentru fiecare client  $C_{ij}$  în CumparatoriPentru( $I_i$ )
            Pentru fiecare produs  $I_k$  în CumprateDe( $C_{ij}$ )
                Asocieri := Asocieri U {(  $I_i$  ,  $I_k$  )}
            Pentru fiecare produs  $I_k$  asociat cu  $I_i$ 
                Similaritati := Similaritati U (  $I_i$  ,  $I_k$  , CalculSimilaritate( $I_i$  ,  $I_k$  ) )
            }
        }
    }

```

Există multe modalități de calcul a similarității între două produse. O metodă des folosită este funcția *cosinus*. Se utilizează câte un vector care corespunde unui produs (din n produse în total), iar cele m elemente ale vectorului corespund notelor acordate de clienți produsului. Similaritatea între două produse se calculează aplicând funcția *cosinus* asupra vectorilor corespunzători produselor. Din această pricină, calculul *offline* al tabeli similarităților între produse implică o complexitate ridicată, $O(n^2*m)$. În practică ea este de fapt mai aproape de $O(n*m)$, pentru că majoritatea clienților cumpără un număr redus de produse. Fiind dată o matrice a produselor similare, algoritmul găsește elemente similare pentru fiecare produs cumpărat de un utilizator pe baza notelor date de utilizator produselor. Acest calcul este mult mai rapid, depinzând doar de numărul de produse cumpărate sau notate de utilizator.

UseNet News

Sistemul UseNet (Konstan et. al, 1997) dorește să ofere recomandări superioare față de ce oferă alte aplicații asemănătoare în domeniul știrilor. Studiul pilot realizat la începutul proiectului a arătat că se poate implementa

un sistem de recomandări pentru compania UseNet News, însă au apărut câteva provocări:

- Integrarea filtrării colaborative într-un sistem informatic ce conține utilizatori deja definiți, aplicații și interfețe și o arhitectură deschisă care suportă mai multe aplicații de știri.
- Adresarea naturii dinamice și distribuite a platformei *UseNet News*. Articolele au o perioadă de viață scurtă și nu există un mediu de stocare central pentru toate articolele de știri.
- Utilizarea unui set de note dispersat.
- Oferirea de performanțe acceptabile pentru utilizator și oferirea unui mecanism de scalare pentru că numărul de utilizatori și articole este în continuă creștere.

Toate acestea trebuie luate în considerare pentru a calcula utilitatea predictivă a proiectului, așa cum este prezentat și în (Konstan et. al, 1997). Această noțiune se referă la valoarea adusă de calculul unei predicții pentru un element. Utilitatea predictivă este o funcție pentru calculul relativ al cantității produselor dorite sau nedorite și calitatea predicțiilor. Nivelul de utilitate al unui obiect este o măsură subiectivă, specifică fiecărui utilizator în parte. Tocmai din această cauză, produsele nu pot fi clasificate drept bune sau proaste în mod global.

Autoritatea unui utilizator în sistemele de recomandare

Hosu și Trăușan-Matu (2007) propun o metodă de ordonare a recomandărilor pe baza unui algoritm similar PageRankului, în contextul unui sistem de recomandare de imagini. Se consideră că un utilizator care deține o colecție de imagini de calitate este mai capabil să aprecieze alte imagini de calitate (este o autoritate în materie). Așadar, un utilizator transmite indirect autoritate mai departe altor utilizatori prin marcarea ca și favorite a unor anumite poze ale acestora. Calitatea unei poze este dată de suma ponderată a autorităților utilizatorilor care au marcat-o ca și favorită. S-a obținut prin folosirea acestui algoritm o ordonare mai bună a pozelor recomandate decât prin criteriul clasic al numărului de favorite exprimate pentru o imagine. Termenul obiectiv de comparație a fost considerat numărul de favorite exprimate pe termen lung pentru acea imagine.

Metode de bootstrapping pentru sisteme de recomandare

Sistemele de recomandare debutează fără informații despre utilizatori sau itemi. (Olsson, Rudström, 2003) propun o metodă care inițializează (eng. *bootstraps*) un sistem de recomandare cu un set de profiluri artificiale de utilizator. Acestea sunt obținute prin eșantionarea unui model probabilistic construit din cunoștințe anterioare. Ei au utilizat această tehnică asupra unui algoritm de recomandare bazat pe K Nearest Neighbors care realizează recomandări pentru filme, obținând o îmbunătățire a performanțelor algoritmului pentru predicții singulare.

2.3 Regăsire de imagini pe baza conținutului

Regăsirea de imagini pe baza conținutului vizual utilizează tehnici de detecție a trăsăturilor vizuale, modalități de exprimare a lor prin intermediul descriptorilor vizuali și metrice de similaritate pe baza descriptorilor care permit identificarea celor mai apropiate imagini din punct de vedere al anumitor aspecte vizuale.

În acest domeniu, s-au căutat cele mai bune practici pentru identificarea conceptelor vizuale din imagini, prin analiza rezultatelor obținute de echipele câștigătoare la ImageClef 2010, LEAR și XRCE (Mensink et al., 2010) și Universitatea din Amsterdam (van der Sande, Gevers, 2010). De acolo, au fost identificați descriptorii SIFT - *Scale Invariant Feature Transform* (Lowe, 2004), alături de clasificarea pe bază de SVM și de necesitatea de a combina descriptorii SIFT cu descriptori de culoare pentru a include atât elemente de textură și formă locală, cât și elemente de culoare în calculul similarităților. Utilizarea tehnicii *bag of visual words* și recurgerea la crearea unui vocabular vizual în aplicația propusă a fost, de asemenea, inspirată de rezultatele deosebite obținute în competiție de cele două echipe și nu numai.

Descriptorul SIFT este folosit cu rezultate foarte bune în acest domeniu pentru extragerea de trăsături care sunt, într-o anumită măsură, invariante la modificări precum intensitatea luminii, zgomotul în imagine, rotație, scalare etc. Potrivit lui (Lowe, 2004), procesul de detecție a acestui tip de trăsături implică:

- *Scale-space extrema detection* = detectarea punctelor de interes reprezentate de puncte de extrem DoG (diferențe-of-Gaussian) la diferite scări.
- Localizarea punctelor cheie = puncte de maxim / minim local pentru DoG; se compară fiecare pixel din imaginea transformată DoG cu cei 8 vecini de la o anumită scală, plus vecinii de la celelalte scale.
- Asignarea orientării pentru fiecare punct cheie. Se bazează pe calculul unei histogramme de gradient de orientare în vecinătatea punctelor cheie.
- Generarea de descriptori pentru punctele cheie. Pe baza orientării punctelor cheie, descriptorul se obține ca un set de histogramme de orientare pe vecinătăți de 4x4 pixeli.

2.4 Regăsire de imagini pe baza metadatelor semantice

Biblioteca RiTa (<http://www.rednoise.org/rita/wordnet/>) este un API Java care oferă acces la ontologia WordNet (Fellbaum, 1998), o bază de informații lexicale pentru limba engleză foarte cuprinzătoare și utilizată în numeroase proiecte ce implică procesare de limbaj natural.

RiTa oferă posibilitatea de a regăsi sinonime, antonime, hipernime și hiponime, dar și holonime, meronime, identificarea celei mai probabile părți de vorbire pentru un cuvânt, nominalizări, suport pentru algoritmul *Soundex*, anagrame, potrivire de sabloane, șamd (Howe, 2009).

De asemenea, RiTa oferă și metrici de distanță între termeni din ontologie, funcționalitate utilizată în prezenta lucrare.

2.5 Mecanisme de clasificare automată de imagini

Support vector machines (SVM) (Cortes, Vapnik, 1995) reprezintă un set de metode de învățare supervizată utilizate în clasificare și regresie. În cele ce urmează, se descrie exclusiv partea de clasificare pe baza SVM. Un clasificator SVM binar se bazează pe un set de antrenare care conține perechi instanță clasă, $(x_i, y_i), i = 1, \dots, l$ unde $x_i \in R$ reprezintă vectorul instanței în spațiul de intrare, iar $y \in \{1, -1\}^l$ reprezintă clasa binară aplicată instanței.

Problema SVM constă în a găsi un hiperplan care să separe în mod optim cele două clase existente în setul de antrenare. Formularea matematică a unui clasificator SVM este reprezentată de următoarea problemă de optimizare:

$$\min_{w,b,\varepsilon} \frac{1}{2} w^T w + C \sum_{i=1}^l \varepsilon_i,$$

Ținând cont de constrângerea:

$$y_i(W^T \Phi(x_i) + b) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0$$

Termenii ε_i reprezintă parametri utilizați pentru gestiunea unei intrări care nu este liniar separabilă și pentru controlarea fenomenului de *overfitting* (degradarea capacității de generalizare, deci de clasificare corectă a unor instanțe noi). C reprezintă parametrul de penalizare pentru termenul eroare ($C > 0$). Funcția Φ este utilizată pentru a mapa vectorii de antrenare într-un spațiu de o dimensiune superioară, în care datele sunt liniar separabile.

$K(x_i, x_j) \equiv \Phi(x_i)^T \Phi(x_j)$ reprezintă funcția *kernel*.

Funcțiile *kernel* cele mai utilizate sunt: *kernel* liniar, polinomial, *radial basis function kernel*, și funcția sigmoidă. Prezentăm în continuare funcția *kernel radial basis function* (RBF):

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0.$$

Un clasificator care utilizează funcția *kernel* RBF are doi parametri care trebuie optimizați, anume C și γ . Optimizarea parametrilor se poate face printr-o abordare de tip *grid search*, o căutare extensivă în anumite intervale alese pentru cei doi parametri. Căutarea are ca scop găsirea perechii de parametri care maximizează acuratețea în setul de testare. Pentru a maximiza potențialul clasificatorului de a gestiona corect exemple necunoscute, se utilizează validarea încrucișată. Validarea încrucișată reprezintă o metodă prin care setul inițial de instanțe se împarte în V părți egale, și la fiecare pas, se realizează antrenarea pe bază a $V-1$ părți din setul de instanțe și testarea pe bază părții rămase.

Clasificarea SVM poate fi făcută în cazul a două clase, cazul unui clasificator binar, pentru o singură clasă (OneClass SVM) – utilizat în detecția anomaliilor, pentru mai multe clase disjuncte (multi-class), unde o

instanță aparține unei clase unice și pentru mai multe clase cu suprapunere, numite etichete (multi-label), unde o instanță poate avea mai multe etichete.

Examinăm două abordări în clasificarea *multi-class*: *one-versus-one* și *one-versus-all*.

Abordarea *one-versus-one* presupune reducerea problemei de clasificare *multi-class* la o suită de probleme de clasificare binară, în care se construiesc $N*(N-1)/2$ clasificatoare binare (unde N reprezintă numărul total de clase). Un clasificator permite antrenarea fiecărei perechi de clase („una contra una”). Pentru stabilirea clasei pentru o instanță dată, se utilizează o metodă de votare, prin care fiecare clasificator acordă un vot clasei pe care o prezice pentru această instanță. Clasa cu cele mai multe voturi este câștigătoare și devine clasa finală prezisă pentru instanță.

Abordarea *one-versus-all* presupune antrenarea a N clasificatori care clasifică o clasă față de toate celelalte clase („una contra restului”). Clasificarea unei instanțe se va baza pe prezicerea acelei clase care maximizează marginea dintre instanță și planul antrenat de separație al clasificatorului.

3 Descrierea sistemului propus

Aplicația are ca scop obținerea de imagini similare dintr-un set numeros, folosind informația semantică asociată imaginilor prin etichetele asiguate de utilizatorii Flickr (www.flickr.com), informația socială definită de relația dintre utilizatori și imaginile similare (pe baza listelor de poze favorite), trăsături vizuale identificate din imaginea propriu-zisă, precum și metode de clasificare automată.

Funcționalitățile incluse în aplicația realizată sunt reprezentate de mecanisme pentru detecția trăsăturilor vizuale din imagini, analiză semantică folosind ontologia WordNet și a bibliotecii Rita, precum și utilizarea platformei Taste din cadrul bibliotecii Apache Mahout ca bază pentru dezvoltarea sistemului de recomandare și pentru realizarea filtrării colaborative. Mai multe detalii despre Taste și modul în care se pot îmbina funcționalitățile acestuia pentru realizarea unui sistem de recomandare personalizat se găsesc în (Adjiman, 2009) și în (Ingersoll, 2009). Alte funcționalități auxiliare sunt reprezentate de generarea de descriptori vizuali pentru stocare locală, indexarea imaginilor și a trăsăturilor descoperite, mecanisme de calcul de similarități între imagini pe baza trăsăturilor,

mecanisme de ordonare pe baza combinării mai multor tipuri de măsuri, precum și mecanisme de clasificare a imaginilor pe baza trăsăturilor vizuale din imagini.

Rezumăm mecanismele de recomandare oferite:

- Vizualizarea recomandărilor bazate pe imagine folosind o combinație a următoarelor aspecte:
 - Social – filtrare colaborativă bazată pe articol;
 - Semantic – conținut semantic folosind etichetele imaginii și metrice de similaritate oferite de RiTa;
 - Vizual – conținut vizual al imaginii și metrice de similaritate vizuală oferite de biblioteca LIRE (Lux, Savvas, 2008), utilizată în cadrul proiectelor Caliph și Emir (Lux, 2009) - două biblioteci Java pentru adnotarea și regăsirea imaginilor folosind adnotări sub formă de grafuri semantice și regăsirea imaginilor pe baza descriptorilor din standardul MPEG-7 (<http://mpeg.chiariglione.org/standards/mpeg-7/mpeg-7.htm>);
 - Clasificare automată – se utilizează clasificatorul SVM pentru a prezice clasa imaginii și a o asocia cu imagini aflate în aceeași clasă.
- Vizualizarea recomandărilor bazate pe profil (filtrare colaborativă bazată pe utilizator)
- Vizualizarea recomandărilor bazate pe context – realizată pentru un utilizator autentificat în momentul în care vizualizează o imagine. Acest tip de recomandare utilizează conținutul imaginii (ținând cont de cele patru aspecte menționate anterior) și filtrare colaborativă bazată pe utilizator.

3.1 Componentele aplicației

Prezentăm în continuare diagrama arhitecturală a aplicației propuse.

Interfața externă a sistemului propus este asigurată de un serviciu web SOAP implementat în Java și rulat utilizând un motor de servicii web Apache Axis2.

În ce privește partea de client, un utilizator poate accesa sistemul printr-un browser având instalat un script dedicat (eng. *userscript*), care realizează

comunicația cu serviciul web expus de sistem. Scriptul, realizat în JavaScript, folosește apeluri Ajax pentru a obține pozele recomandate într-un anumit context, pagina Flickr vizualizată, și pentru a le integra în pagină într-un mod transparent pentru utilizator.

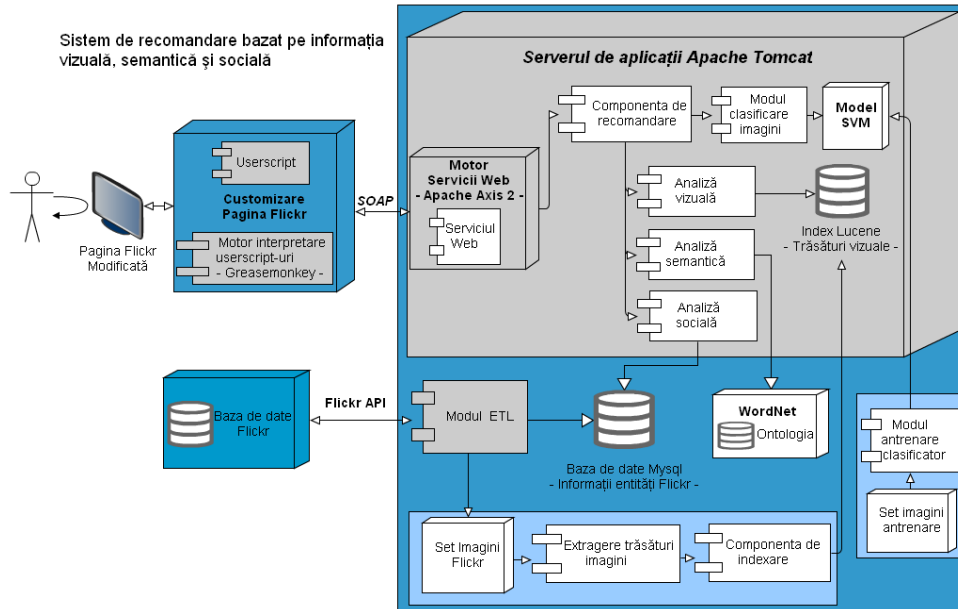


Figura 1. Arhitectura aplicației propuse

De asemenea, componentele pe partea de server ale aplicației sunt găzduite de serverul de aplicații Apache Tomcat 6. Detaliile despre aceste componente sunt oferite în continuare.

3.1.1 Modulul ETL și baza de date de imagini

Pentru a demonstra funcționalitatea aplicației noastre am ales ca bază de obținere a datelor necesare aplicația Flickr. Obținerea informațiilor și salvarea lor local s-a realizat folosind un crawler web inițializat cu pozele încărcate în Flickr într-o zi dată și care folosește relațiile între utilizatori și imagini, alternând extragerea utilizatorilor care au preferat pozele existente cu pozele preferate de utilizatorii extrași, cu scopul de a completa cât mai eficient profilurile de preferințe ale utilizatorilor. Informațiile legate de utilizatori, lista lor de contacte, pozele încărcate, pozele favorite, poze

încărcate, etichete asociate sunt extrase de acest *crawler*, transformate și încărcate într-o bază de date MySQL.

Pe lângă informațiile sociale menționate anterior, s-au salvat local imaginile pentru realizarea extragerii trăsăturilor vizuale. Pentru aceasta, există o componentă care extrage câte un identificator de imagine din baza de date, obține URL-ul imaginii din Flickr și descarcă imaginea.

3.1.2 Extragerea trăsăturilor vizuale din imagine

Pentru extragerea trăsăturilor din imagine, am folosit metodele implementate deja în sistemul LIRE, astfel avem indecși creați pentru ColorLayout, ColorHistogram, SIFT, histograme bazate pe SIFT. Biblioteca LIRE conține o gamă largă de descriptori ce pot fi folosiți pentru a obține o informații cât mai detaliate privind elementele din imagine. Numeroși descriptori vizuali sunt descriși și în lucrarea (Deselaers, Keysers, Ney, 2007).

Pe baza descriptorului SIFT se crează mai departe un model de tipul „*visual words*”. Descriptorii SIFT sunt separați în clustere folosind un algoritm K-Means, astfel obținându-se un vocabular de K clustere, similare unor „termeni vizuali”. Imaginea este apoi descrisă printr-o histogramă a frecvenței cu k elemente (*bins*), câte unul pentru fiecare cluster.

De asemenea, a fost implementat un algoritm de „piramidă spațială”, descris în (Lazebnik, Schmid, Ponce, 2006) pentru a obține o descriere spațială a trăsăturilor SIFT descoperite în imagine. Acest algoritm poate fi folosit și pentru a descoperi forme în imagini, utilizând detectorul de muchii Canny, așa cum este prezentat în (Boch, Zisserman, Munoz, 2007).

Un alt descriptor folosit este ColorLayout. Acesta este inclus în standardul MPEG-7 și descrie culorile dintr-o imagine și localizarea acestora. De asemenea, un descriptor bazat pe culoare utilizat este descriptorul uzual histograma de culoare, pe baza spațiului de culori RGB.

3.1.3 Componenta de clasificare SVM

Pentru predicția clasei unei anumite imagini, s-a realizat antrenarea unui clasificator SVM. Clasificatorul se bazează pe un set de imagini de antrenare reprezentate printr-un set de descriptori vizuali, pe de o parte, și pe informații despre clasa asociată fiecărei imagini.

S-a utilizat un clasificator SVM *multi-class*, cu kernelul *radial basis function*. Pentru realizarea proceselor de antrenare și predicție s-a recurs la biblioteca LIBSVM. Mai multe detalii legate de elementele conținute în librărie sunt prezentate în (Chang, Lin, 2001). Parametrii C și γ au fost optimizați printr-o căutare de tip *grid search* și validare încrucișată, folosindu-se utilitarul de selecție a parametrilor oferit de LIBSVM. În ce privește abordarea problemei multi-class, s-a utilizat metoda *one-versus-one* pusă la dispoziție de LIBSVM și o implementare proprie a metodei de antrenare pentru clasificarea *one-versus-all*.

Componenta auxiliară de antrenare a clasificatorului, care realizează pregătirea și formatarea setului de date de antrenare și de testare, a fost realizată în Java. Componenta extrage descriptorii vizuali din indecșii Lucene (<http://lucene.apache.org>) creați și îi stochează într-un fișier cu formatul compatibil LIBSVM. Pentru antrenarea și testarea clasificatorului, a fost utilizat setul de date MIRFlickr (Huiskes, Lew, 2008), un set folosit și în cadrul competiției internaționale ImageClef, disponibil la adresa <http://press.liacs.nl/mirflickr/>. MIRFlickr conține un set de un milion de imagini descărcate din Flickr, împreună cu etichetele și cu metadatele EXIF (mai multe detalii la <http://www.exif.org>), și adnotate manual cu un set consistent de etichete. Componenta de antrenare realizează și accesul la setul utilizat din MIRFlickr și realizează procesarea adnotărilor oferite pentru acestea și selecția subsetului de imagini pentru antrenare și testare.

3.1.4 Componenta de indexare

Componentele de extragere de trăsături vizuale din imagine și cea de indexare sunt executate secvențial. Metadatele calculate pentru fiecare imagine în parte sunt salvate în indecși Lucene, integrați în proiectul LIRE. Indexul rezultat conține informații pentru descriptorii menționați anterior, plus pentru descriptorul implementat de noi pentru localizarea trăsăturilor SIFT.

În cadrul acestei componente, se realizează calculul și indexarea trăsăturii „*Spatial Pyramid*” menționată mai devreme. Alegerea acestei trăsături s-a bazat pe faptul că include informații despre localizarea trăsăturilor în imagine, astfel permițând realizarea calculelor de similaritate între imagini ținând cont de aspecte de asemănare locală, în anumite regiuni ale imaginii. Piramida implementată are trei niveluri de adâncime, obținând

astfel un grad mai mare de detaliu decât în cazul calculării histogramei clasice de trăsături SIFT.

Pașii algoritmului de extragere a descriptorului de piramida spațială sunt:

1. Calculul trăsăturilor SIFT;
2. Aplicarea unui algoritm de clusterizare pentru delimitarea trăsăturilor în clustere;
3. Calculul histogramei pentru nivelul 0, care constă în reprezentarea numărului de apariții al fiecărui cluster în parte;
4. Împărțirea imaginii în 4 părți egale și calculul histogramei pentru fiecare cadran, luând în calcul trăsăturile ce sunt localizate în interiorul acestuia;
5. Împărțirea fiecărui cadran în alte 4 părți egale și repetarea procedurii până la atingerea adâncimii stabilite;
6. Stocarea într-un index a piramidei conținând histogramele rezultate, sub forma unui vector de dimensiune $M \frac{1}{3}(4^{L+1} - 1)$, unde M reprezintă numărul de trăsături extrase și L reprezintă indicele ultimului nivel (L=2). Înainte de salvare, în vector avem: histograma de la nivelul 0 pe primele M poziții, de la poziția M+1 la 4 * M avem histograma de la nivelul 1 și în continuare histograma de la nivelul 2.

Extragerea trăsăturilor SIFT și calculul clusterelor sunt preluate din librăria LIRE.

3.1.5 Componenta de recomandare

Ca și în cazul extragerii de trăsături și indexării acestora am folosit pentru recomandare și algoritmi deja existenți.

Analiza socială

Pentru partea de analiză socială, realizată prin aplicarea filtrării colaborative, sistemul propus utilizează coeficientul Tanimoto, care reprezintă coeficientul Jaccard extins pentru vectori cu componente binare (detalii în Tan și Steinbach, 2005). În cadrul bibliotecii Taste, coeficientul Tanimoto se folosește pentru calculul de preferințe booleene (valoarea binară constând în prezența / absența preferinței exprimate pentru articol), prin următoarea formulă, unde A și B reprezintă vectorii de preferințe în

spațiul articolelor, care au fiecare N elemente binare, unde N este numărul total de articole:

$$T(A, B) = \frac{A \cdot B}{\|A\|^2 + \|B\|^2 - A \cdot B}$$

Platforma de recomandare a fost realizată prin intermediul extinderii facilităților oferite de biblioteca Taste, mai exact, prin derivarea de noi clase cu funcționalitate specifică pentru clase de recomandare și similarități de articole.

Analiza vizuală

Pentru obținerea de recomandări pe baza piramidei spațiale SIFT am utilizat o metodă de calcul al similarității pentru obținerea celor mai bune rezultate.

Similaritatea la nivel de piramidă este văzută ca suma similarităților pentru fiecare trăsătură în parte ce a fost extrasă din imagine:

$$K^L(X, Y) = \sum_{m=1}^M k^L(X_m, Y_m)$$

Similaritatea între două imagini X și Y este definită ca o suma ponderată a similarității pe fiecare nivel al piramidei:

$$K^L(X, Y) = I^L + \sum_{l=0}^{L-1} \frac{1}{2^{L-l}} (I^l - I^{l+1}) = \frac{1}{2^L} I^0 + \sum_{l=1}^L \frac{1}{2^{L-l+1}} I^l, \text{ unde}$$

ponderea fiecărui nivel este calculată prin $\frac{1}{2^{L-l}}$, fiind invers proporțională cu dimensiunea (lățimea) cadranelor de la acel nivel.

În final, similaritatea pentru o trăsătură pentru un nivel este calculată ca suma minimului de apariții a clusterului din care face parte acea trăsătură.

$$I(H^l_X, H^l_Y) = \sum_{i=1}^D \min(H^l_X(i), H^l_Y(i))$$

Se calculează această similaritate între poza din interogare și celelalte din index și se vor afișa utilizatorului un număr prestabilit de imagini în ordinea relevanței, adică în ordine descrescătoare a similarităților.

Analiza semantică

Pentru recomandarea semantică se folosesc etichetele asociate imaginii de referință și a celorlalte și se calculează gradul de asemănare între acestea.

Distanța semantică între etichete este oferită de biblioteca RiTa care, la rândul ei, folosește ontologia WordNet pentru a avea acces la termeni și la legăturile dintre ei. Măsura finală între etichetele asociate imaginii din interogare T_1 și a unei imaginii T_2 este

$$\frac{1}{|T_1| * |T_2|} \sum_{t_i \in T_1, t_j \in T_2} \max(sim(t_i, t_j)),$$

unde t_1 aparține mulțimii etichetelor asociate imaginii originale, iar t_2 aparține mulțimii etichetelor imaginilor candidate a fi recomandate.

Distanța între doi termeni, $dist(t1, t2)$, inversa similarității, este calculată folosind RiTa. Această metodă folosește „arborele” sub care sunt reprezentate cuvintele în WordNet, considerând nodurile ca reprezentând conceptele, cuvintele, iar arcele reprezintă relațiile definite în ontologie.

Procesul de ordonare

Acest proces presupune ordonarea imaginilor în ordine descrescătoare a similarității față de imaginea de interogare.

Similaritatea pe baza descriptorilor vizuali se realizează prin mecanisme de calcul a distanței oferite în cadrul LIRE între imaginea de interogare (q) și imaginea testată (i). Similaritatea este reprezentată de inversa distanței, precum în ecuația următoare (adăugarea unei unități la numitor evită diviziunea cu zero):

$$Score_{visual}(q, i) = \frac{1}{dist_{visual}(q, i) + 1}$$

Similaritatea pe baza categoriei (clasei) imaginilor este o funcție binară de similaritate, care ia valoarea 1 când clasele celor 2 imagini coincid și 0 altfel. În condițiile în care clasa nu se cunoaște pentru imaginea i , ea este prezisă cu ajutorul clasificatorului, altfel se folosește clasa cunoscută.

După ce se determină atât similaritățile pe baza vizuală, socială și semantică, precum și pe baza categoriei, se utilizează un mecanism de ponderare a acestora. Formula utilizată în ordonare este o formulă ponderată, cu ponderi stabilite experimental. Scorul final al similarității între două poze q și i este dat de formula următoare, iar astfel imaginile cu cele mai mari scoruri sunt întoarse ca imagini recomandate.

$$Score(q, i) = w_{viz} * Score_{viz}(q, i) + w_{soc} * Score_{soc}(q, i) \\ + w_{sem} * Score_{sem}(q, i) + w_{clasif} * Score_{clasif}(q, i)$$

3.2 Rezultate experimentale

3.2.1 Rezultatele componentei de clasificare

Subsetul din setul de date MIRFlickr care a fost utilizat pentru antrenarea și testarea clasificatorului SVM include aproximativ 400 de imagini, cuprinzând imagini din 4 clase distincte: *tree*, *night*, *sea*, *flower*, fiind aproximativ egal reprezentate în setul de date. S-a realizat o variantă de clasificare *multi-class*, bazată pe abordarea *one-versus-one* și o variantă de clasificare multi-label, bazată pe aplicarea abordării *one-versus-all* și cumulara etichetelor rezultatele.

Table 1. Sumarul performanțelor celor patru experimente de clasificare *one-versus-one*

Nr. Crt.	Trăsături	Nr Atribute	Parametrii optimi ai clasificatorului	Acuratețe la testare
1	Histograma SIFT cu 32 bins	32	C = 32768.0 $\gamma = 0.0078125$	41.1%
2	ColourLayout	118	C = 32.0, $\gamma = 0.000488$	51.6%
3	ColourHistogram	512	C = 0.03125, $\gamma = 0.00781$	25.6%
4	ColorLayout + Histograma SIFT (32 bins)	150 (118+32)	C = 512.0, $\gamma = 3.05e-05$	53.3%

În varianta *one-versus-one*, s-a extras un set de imagini de circa 400 de imagini, cu condiția ca o imagine să nu aibă decât una dintre cele 4 etichete, câte 100 de imagini pentru fiecare clasă. Subsetul de date a fost împărțit în set de date de antrenare (75%) și set de date de testare (25%). S-au indexat imaginile din setul de antrenare și cel de testare și s-a generat fișierul de intrare în formatul compatibil cu LIBSVM. Antrenarea a fost realizată prin instrumentul de selecție automată a parametrilor oferit de LIBSVM, utilizând scalarea automată a datelor de intrare și validarea încrucișată. Sumarul rezultatelor clasificării prin cele patru experimente *one-versus-one*

este prezentat în tabelul 1, evidențiindu-se numărul de atribute utilizate în clasificare, precum și cei doi parametri optimi ai clasificatorului – C , factorul de penalizare și γ , parametrul *kernel*-ului RBF.

Metoda *one-versus-all* a fost a doua variantă de clasificare utilizată, în care se antrenează câte un clasificator pentru fiecare clasă, care să discearnă între instanțele aparținând clasei și instanțele celorlalte clase. Pentru o instanță necunoscută, clasa finală este dată de clasificatorul cu răspunsul cel mai puternic pe baza funcției de decizie. Am utilizat conceptul *one-against-all* pentru a realiza un experiment *multi-label* în care o imagine primește etichete doar de la clasificatorii care clasifică instanța ca instanță pozitivă pentru acea etichetă.

Pentru a compara performanțele obținute prin cele două abordări, s-a utilizat tot un set de aproximativ 400 de imagini, care au avut 1-2 etichete din cele 4 (*tree, night, sea, flower*) și s-au creat 4 fișiere de intrare LIBSVM, câte unul pentru fiecare etichetă. Am folosit trăsăturile vizuale (Color Layout și histograma SIFT) din experimentul cel mai bun (nr. 4) obținut în abordarea *multi-class one-versus-one*. Rezultatele experimentului sunt redade în continuare.

Tabel 2. Sumarul performanțelor experimentului *one-versus-all*

Nr. Crt.	Clasa	Rezultat	Acuratețe la testare
1	Flower	$C=2.0, \gamma=0.03125$, acuratețe la antrenare = 75.7%	74.7% (71/95)
2	Night	$C=8.0, \gamma=0.0078125$, acuratețe la antrenare = 77.0%	75.8% (72/95)
3	Sea	$C=2.0, \gamma=0.03125$, acuratețe la antrenare = 81.0%	80.0% (76/95)
4	Tree	$C=0.03125, \gamma=0.0078125$, acuratețe la antrenare = 73.3%	76.8% (73/95)

Din tabelul nr. 2, se observă că prin abordarea *one-versus-all* s-a obținut o acuratețe mai bună decât cea obținută în experimentul nr. 4 din abordarea *one-versus-one* (tabelul 1). S-au obținut patru modele SVM distincte, fiecare constând într-un clasificator binar pentru o anumită clasă. Se poate propune

un algoritm de predicție pentru a îndeplini sarcina de adnotare *multi-label*, anume realizarea predicției prin intermediul celor patru clasificatoare binare.

În urma experimentelor realizate, s-a observat că descriptorul de distribuție spațială a culorii (ColorLayout) a avut rezultate mai bune în clasificare, singur, decât experimentul bazat doar pe descriptori SIFT, lucru care se explică prin uniformitatea imaginilor din anumite clase (ex: *night*, *sea*). Cele mai bune rezultate au fost obținute prin combinarea descriptorului de distribuție spațială a culorii cu histogramele SIFT. Această combinație permite alăturarea informației despre puncte de interes SIFT cu informația de culoare, care nu este surprinsă în descriptorul SIFT. Descriptorul histogramă de culoare în spațiul RGB, un descriptor foarte comun, a avut cele mai proaste performanțe. Acest rezultat își găsește explicația în faptul că descriptorul nu ține cont de distribuția spațială a culorilor în imagine și realizează astfel o reprezentare globală a imaginii, fără localizarea culorii.

Experimentul de clasificare *one-versus-all* a fost mult mai performant decât cel mai bun experiment *one-versus-one*, lucru ce se justifică prin faptul că, în abordarea *one-versus-all*, clasificatorii înregistrează diferențele dintre o clasă țintă și toate celelalte clase, în vreme ce în abordarea *one-versus-one*, clasificatorii fac discernerea doar între 2 clase, pierzând din performanță deoarece nu izolează factorii global discriminativi pentru o clasă dată.

3.2.2 Rezultatele recomandării

În ce privește sistemul de recomandare, s-a utilizat un set de date extras din Flickr pe baza API-ului Flickr și a modulului ETL. Setul de date extras conține circa 3000 de imagini, pentru care în baza de date au fost introduse informații despre etichetele imaginilor, preferințele exprimate de utilizatori asupra acestora și informații despre acești utilizatori, împreună cu informații despre cei care au încărcat imaginile.

Pentru stocarea trăsăturilor vizuale, s-a realizat un index Lucene conținând descriptorii: ColorHistogram (RGB), ColorLayout, trăsături SIFT, histograma SIFT prin clusterizare folosind K-Means, descriptorul de piramidă spațială pe baza de SIFT (eng. *spatial pyramid matching*).

În tabelul 3 sunt prezentate exemple de recomandări în funcție de diverse criterii. Rezultatele obținute doar pe baza criteriilor semantice au fost imagini similare la nivel conceptual și mai puțin ca aspect vizual general. Recomandarea bazată pe piramida SIFT a produs imagini cu elemente locale

asemănătoare, dar cu similaritate redusă la nivel global. Recomandările care au combinat distribuția spațială a culorii cu filtrarea colaborativă au produs imagini mai asemănătoare la nivel global cu imaginea sursa. În sfârșit, recomandările care au combinat toate trăsăturile disponibile - social, semantic, distribuție spațială a culorii și clasificator de imagini pe baza trăsăturilor vizuale - au obținut în general imagini cu temă conceptuală și vizuală similară. Introducerea criteriului social a promovat în lista recomandărilor și imagini mai puțin asemănătoare vizual, dar prezise ca interesante pentru utilizator, pe baza procesării preferințelor altor utilizatori din sistem.

Tabel 3. Exemple de recomandări realizate



S-a putut remarca experimental că detecția tipului de scenă dintr-o imagine duce la furnizarea de recomandări mai credibile, datorat faptului că se încearcă recomandarea de imagini din aceeași clasă ca și imaginea sursă. În timp ce recomandarea pe baza informațiilor de nivel jos, pur vizuale, nu

implică nici o evaluare a înțelesului scenei, ci doar a componentelor tehnice (culoare, formă, textură), recomandarea care se bazează și pe clasa imaginii aduce o informație semantică prețioasă.

Astfel, pornindu-se de la o imagine de interogare, sistemul a comparat clasa prezisă pentru imaginea sursă cu clasele (prezise sau cunoscute) ale imaginilor candidate pentru recomandare și a selectat imaginile cele mai similare vizual, social și semantic, dar din aceeași clasă cu imaginea sursă. Scorurile sociale, vizuale și semantice, alături de coinciderea clasei imaginii sursă cu imaginea candidată, se calculează independent și se ponderează într-o similaritate finală. Rezultatele obținute astfel urmăresc așadar mai multe criterii pentru a obține imagini potrivite cu interesul de moment al utilizatorului și interesul general, dat de profilul de preferințe pe termen lung.

În ce privește evaluarea performanțelor componente de recomandare implementate, o evaluare cantitativă a componente este îngreunată semnificativ de caracterul particular al preferințelor pentru imagini din Flickr - de tip *boolean*, caz în care doar existența preferinței pentru o poză este indicată, iar absența preferinței poate exprima în mod egal fie un vot negativ, fie că utilizatorul nu a văzut sau evaluat imaginea respectivă. Acest fapt impune mari dificultăți în găsirea unei metode pertinente de evaluare. O metodă clasică în ce privește sistemele de recomandare presupune eliminarea unor N preferințe dintre cele cunoscute pentru un utilizator și verificarea faptului că, dacă aceste preferințe se regăsesc apoi în cele mai bune N recomandări făcute de sistem pe baza preferințelor neeliminate. Însă, având în vedere cantitatea încă limitată de date și numărul mic de preferințe existente per utilizator în cadrul setului de date de test extras pentru platforma demonstrativă a sistemului propus, se așteaptă ca utilizarea unei astfel de metode să furnizeze rezultate neconcludente.

Etapă următoare a dezvoltării sistemului are în vedere obținerea unui set mai cuprinzător de date și explorarea și determinarea unor metrici de evaluare cantitativă eficiente pentru aceste tipuri de sisteme de recomandare.

4. Direcții de dezvoltare viitoare

Pentru îmbunătățirea rezultatelor se poate încerca în faza de preprocesare a imaginilor extragerea de descriptori SIFT la scări multiple. Tot în aceeași fază, se poate încerca extragerea de trăsături prin „*dense sampling*”,

constând în selecția unei mostre din imagine la fiecare X pixeli distanță. Această metodă ar îmbunătăți recunoașterea de imagini uniforme din care nu se pot extrage multe puncte de interes. Utilizarea unui descriptor de culoare local îmbinat cu *dense sampling*, poate îmbunătăți performanțele clasificatorului.

Gruparea utilizatorilor prin metode de clustering pentru a optimiza viteza de calcul al recomandărilor în cazul filtrării colaborative este văzută de asemenea ca o posibilă îmbunătățire. Adăugarea de transparență în recomandarea dată utilizatorului, pe lângă imaginile recomandate, care să prezinte utilizatorului mai multe informații referitoare la motivația pentru care o anumită recomandare i-a fost oferită. Dezvoltarea unui sistem prin care utilizatorul să poată da *feedback* pentru recomandările primite și sistemul să se adapteze la aceste elemente este un alt aspect care poate aduce îmbunătățiri.

5. Concluzii

Lucrarea trece în revistă starea curentă de dezvoltare a domeniului de regăsire de imagini în funcție de conținutul vizual și prezintă o adaptare a acestei probleme intens documentate în literatura de specialitate în alt domeniu de interes, apropiat, reprezentat de sistemele de recomandare.

În cadrul aplicației, au fost analizate cele două mari categorii de recomandări, anume recomandări bazate pe conținut, respectiv prin filtrare colaborativă. Au fost studiate măsuri de similaritate specifice fiecărui tip de analiză și au fost rafinate pentru a implementa o aplicație de recomandare de imagini, plecând de la sistemul Flickr. Aplicația propusă oferă utilizatorilor Flickr patru operații de recomandare de bază: similaritate vizuală, folosind descriptorul de culoare ColorLayout împreună cu o soluție implementată de noi „Spatial Pyramid”, tehnici de recomandare semantică, folosind etichetele asociate imaginii și o bibliotecă peste ontologia WordNet pentru a aplica o măsură de similaritate între etichete, recomandare folosind filtrarea colaborativă, cu ajutorul bibliotecii Taste și de asemenea folosirea unui clasificator pentru obținerea de imagini din aceeași categorie pe baza elementelor vizuale.

În ce privește recomandarea pe bază de conținut vizual, s-au căutat cele mai bune practici pentru identificarea conceptelor vizuale din imagini prin

studiul rezultatelor înregistrate la competițiile ImageClef 2010 și 2009. De acolo, au fost identificați descriptorii SIFT, alături de clasificarea pe bază de SVM și de necesitatea de a combina descriptorii SIFT cu descriptori de culoare pentru a include atât elemente de textură și formă locală, cât și elemente de culoare în calculul similarităților. Utilizarea tehnicii *bag of visual words* și recurgerea la crearea unui vocabular vizual în aplicația propusă a fost, de asemenea, inspirată de rezultatele deosebite obținute de LEAR și XRCE și Universitatea din Amsterdam.

În concluzie, prin studiile realizate și implementarea aplicației propuse, am putut evalua beneficiile și, mai ales, utilitatea deosebită a unui sistem de recomandare, în condițiile în care spațiul de căutare este foarte vast și greu de explorat pentru utilizator.

Referințe

- Adjjiman P. (2009). Flexible Collaborative Filtering in JAVA with Mahout Taste. Descărcat de la <http://philippeadjiman.com/blog/2009/11/11/flexible-collaborative-filtering-in-java-with-mahout-taste/> la 01.02.2011.
- Boch A, Zisserman A, Munoz X. (2007). Representing shape with a spatial pyramid kernel. CIVR'07, Iulie 2007, Amsterdam, Olanda. Descărcat de la <http://eprints.pascal-network.org/archive/00003009/01/bosch07.pdf>
- Chang C.-C., Lin C.-J. (2001). LIBSVM : a library for support vector machines. Biblioteca LIBSVM disponibilă la <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
- Cortes C., Vapnik V. (1995). Support-Vector Networks, Machine Learning, Vol. 20, Nr. 3, pg. 273-297, DOI: 10.1007/BF00994018.
- Deselaers T, Keysers D, Ney H. (2007). Features for Image Retrieval: An Experimental Comparison. Descărcat de la http://thomas.deselaers.de/publications/papers/deselaers_infret08.pdf
- Fellbaum C. et al (1998). Wordnet: an electronic lexical database. În *Proceedings of International Conference on Computational Linguistics*.
- Hosu V., Trausan-Matu S. (2007). Metodă de recomandare bazată pe rang, considerând satisfacția utilizatorilor, în D.M.Popovici, A.Marhan (eds.), *Proceedings la a 4-a Conferință Națională de Interacțiune Om-Calculator – RoCHI 2007, MATRIX ROM, București, 2007*, pp.129-132
- Howe D. (2009). RiTa: creativity support for computational literature. În *Proceeding of the seventh ACM conference on Creativity and cognition 2009*.
- Huiskes M. J., Lew M. S. (2008). The MIR Flickr Retrieval Evaluation. ACM International Conference on Multimedia Information Retrieval (MIR'08), Vancouver, Canada. Disponibil la <http://press.liacs.nl/mirflickr/mirflickr.pdf>.
- Ingersoll G. (2009). Introducing Apache Mahout. Disponibil la <https://www.ibm.com/developerworks/java/library/j-mahout/index.html>
- Konstan J.A., Miller B.N., Maltz D., Herlocker J.L., Gordon L.R., Riedl J. (1997).

- GroupLens: Applying Collaborative Filtering to Usenet News.
- Lazebnik S, Schmid C, Ponce J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. În *Proc. of CVBR 2006*. Disponibil la <http://www.di.ens.fr/willow/pdfs/cvpr06b.pdf>
- Linden G. (2009). What is a Good Recommendation Algorithm? În *Communications of the ACM*. Disponibil la <http://cacm.acm.org/blogs/blog-cacm/22925-what-is-a-good-recommendation-algorithm/fulltext>
- Linden G., Smith B., York J. (2003). Amazon.com Recommendations: Item-to-Item Collaborative Filtering. În *IEEE Internet Computing*, vol. 7, no. 1, pp. 76-80, Ian./Feb. 2003, doi:10.1109/MIC.2003.1167344
- Lowe D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. În *International Journal of Computer Vision*, 2004. DOI: 10.1.1.157.3843
- Lux M. (2009). Caliph & Emir: MPEG-7 photo annotation and retrieval. În *Proceedings of the seventeen ACM international conference on Multimedia*, pg. 925-926, 2009, Beijing, China. <http://www.semanticmetadata.net/features/>
- Lux M, Savvas C. (2008). LIRE: Lucene Image Retrieval – An Extensible Java CBIR Library. În *Proceedings of the 16th ACM International Conference on Multimedia*, pp. 1085-1088, Vancouver, Canada. , 2008. Descărcat de la <http://www.semanticmetadata.net/lire/>
- MacManus R. (2009). Top 5 Web Trends of 2009: Personalization. Disponibil la http://www.readwriteweb.com/archives/top_5_web_trends_of_2009_personalization.php
- Mensink T., Csurka G., Perronn F., Sanchez J., Verbeek J. (2010). - LEAR and XRCE's participation to Visual Concept Detection Task - ImageCLEF 2010. Descărcată de la http://clef2010.org/resources/proceedings/clef2010labs_submission_65.pdf la 1.02.2011
- Olsson T., Rudström Å. Genesis: A method for bootstrapping recommender systems using prior knowledge. Apărută în Olsson T. (2003). Bootstrapping and Decentralizing Recommender Systems - IT Licentiate theses, 2003-006. Uppsala University, Department of Information Technology, Uppsala, Suedia.
- Resnick P., Varian H. (1997). Recommender systems. În *Communications of the ACM*, Volume 40, Issue 3 (March 1997). Pg: 56 – 58.
- John Riedl J, Todd Beaupre T, Jon Sanders J. (2009) - Research Challenges in Recommenders. Prezentare la *ACM Recommender System 2009*. Descărcată la 01.02.2011 de la <http://recsys.acm.org/2009/tutorial3.pdf>
- Sarwar B, Karypis G, Konstan J, 2001. Item-Based Collaborative Filtering Recommendation Algorithms. În *WWW10*, Mai 2001 , Hong Kong.
- Shani G, Gunawardana A. (2011). În *Recommender Systems Handbook 2011*, Part 2, 257-297, DOI: 10.1007/978-0-387-85820-3_8.
- Tan, P.-N., Steinbach, M., Kumar, V. (2005). Introduction to Data Mining, Addison Wesley.
- van de Sande K, Gevers T. (2010). The University of Amsterdam's Concept Detection System at ImageCLEF 2010. Descărcat de la http://clef2010.org/resources/proceedings/clef2010labs_submission_101.pdf la

01.02.2011.

- Vermorel J., Mohri M. (2004). Multi-Armed Bandit Algorithms and Empirical Evaluation. În *European Conference on Machine Learning*. Descărcat de la <http://www.cs.nyu.edu/~mohri/pub/bandit.pdf> la 01.02.2011.
- Wang J., de Vries A., Reinders M. (2006). Unifying Userbased and Itembased Collaborative Filtering Approaches by Similarity Fusion. În *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.