**Clasificarea Mesajelor**

**Modulul Categorii**

**1.Descriere**

Acest modul se ocupa de definirea categoriilor si clasificarea mesajelor. Scopul nostru este de a implementa un mecanism cu ajutorul caruia separam categorii. Punem la dispozitia userului trei optiuni:

* Definirea unei categorii pe baza unor indicii (o lista de keywords, exemple de texte, pattern-uri etc.)
* Clasificarea automata a textelor primite de la utilizator in functie de categoriile definite anterior.
* Clusterizarea, gruparea unor texte pe baza unor trasaturi comune.

**2.Domenii**

Se vor descrie scenariile de utilizare si modul de interactiune a clientului cu aplicatia.

**3.Scenarii de utilizare**

**3.1 Clientul doreste sa defineasca o categorie**

**3.1.1 Obiectiv/Context**

O categorie va fi definita pe baza indiciilor oferite de utilizator.

**3.1.2 Scenariu/Pasi**

1. Utilizatorul acceseaza aplicatia

2.Acesta efectueaza logarea

3.Aplicatia afiseaza meniul Home

4. Userul selecteaza optiunea de definire a unei categorii

5.Aplicatia afiseaza formularul pentru definirea unei categorii

6. Userul introduce numele categoriei

7.Aplicatia ii prezinta o lista de modalitati de definire a unei categorii precum : adaugarea unei liste de keywords, adaugarea unor texte model, setarea limbii, adaugarea unor pattern-uri etc.

8.Userul selecteaza cel putin una dintre optiunile prezentate anterior

9. Aplicatia ii cere acestuia sa introduca detaliile necesare in functie de optiunile alese

10. Userul este instiintat printr-un mesaj de confirmare de catre aplicatie ca s-a salvat categoria

11. Aplicatia afiseaza din nou meniul Home.

**3.2 Clientul doreste gruparea automata a unor texte pe baza categoriilor predefinite**

**3.2.1 Obiectiv/Context**

Pentru fiecare text introdus, aplicatia afiseaza categoriile din care poate face parte acel text.

**3.2.2 Scenariu/Pasi**

1. Utilizatorul acceseaza aplicatia

2.Acesta efectueaza logarea

3.Aplicatia afiseaza meniul Home

4. Userul selecteaza optiunea de grupare automata

5. Aplicatia verifica daca au fost definite anterior categorii

5.1. In cazul in care nu au fost predefinite categorii, clientul este redirectat spre pagina de adaugare a unor categorii

6.Aplicatia afiseaza o fereastra de upload a textelor

7. Userul incarca textele dorite din fisier/arhiva

8.Aplicatia afiseaza categoriile de care apartine fiecare text, daca s-a gasit macar unul.

**3.3 Clusterizare-gruparea unor texte pe baza unor trasaturi comune**

**3.3.1 Obiectiv/Context**

Gruparea textelor oferite de utilizator pe baza gradului de similitudine dintre ele.

**3.3.2 Scenariu/Pasi**

1. Utilizatorul acceseaza aplicatia

2.Acesta efectueaza logarea

3.Aplicatia afiseaza meniul Home

4. Userul selecteaza optiunea de clusterizare

5.Aplicatia afiseaza o fereastra de upload a textelor

6. Userul incarca textele dorite din fisier/arhiva

7.Aplicatia afiseaza textele grupate pe posibile categorii

**4.State of the art si resurse utile**

-Pentru detectarea limbii: Am putea folosi **Apache Tika** <http://tika.apache.org/1.5/detection.html#Language_Detection>

-Clusterizare si definirea unor categorii: **algoritmul Naïve Bayes** <https://en.wikipedia.org/wiki/Naive_Bayes_classifier>

Am putea folosi algoritmul Naïve Bayes pentru clasificarea textelor, grupandu-le in clustere definite algoritmic. Aceasta metoda este utila pentru cuantificarea unor cantitati mari de informatie, fiecare grupa(cluster) reprezentand mai multe puncte avand caracteristici similare (de exemplu, toate textele care au in componenta informatii despre programarea unui examen vor fi puse in aceeasi grupa).

k-clustering

<http://java-ml.sourceforge.net/api/0.1.7/net/sf/javaml/clustering/KMeans.html>

<http://www.dataonfocus.com/k-means-clustering-java-code/>

**Weka** (Data mining software in Java)

Weka este o colectie de algoritmi de invatare pentru data mining. Algoritmii pot fi aplicati fie direct pe un set de date sau folositi chiar din codul Java. Weka contine instrumente pentru preprocesarea datelor, clasificare, regresie, reguli de asociere si pentru vizualizare. De asemenea, este potrivit pentru dezvoltarea de noi scheme de invatare.

Weka este un software open source, sub licenta generala publica GNU.

In afara schemelor actuale de invatare, WEKA contine, de asemenea, o varietate mare de instrumente care pot fi folosite pentru preprocesarea seturilor de date, astfel incat este posibila concentrarea pe propriul algoritm fara a se tine cont de detalii precum citirea datelor din fisier, implementarea algoritmilor de filtrare si oferirea de code sursa pentru evaluarea rezultatelor.

<http://www.cs.waikato.ac.nz/ml/weka/>

<http://www.inf.ed.ac.uk/teaching/courses/dme/html/Tutorial.pdf>

<https://weka.wikispaces.com/Text+categorization+with+WEKA>

Formatul de input pentru text categorization este ARFF (o librarie utila pentru transformarea documentelor txt in arff <https://weka.wikispaces.com/ARFF+files+from+Text+Collections>)

Filmulete care explica modul de utilizare a librariei WEKA

<https://www.youtube.com/watch?v=Xyl2csNfDsA&index=19&list=PLea0WJq13cnBVfsPVNyRAus2NK-KhCuzJ>

Weka poate fi utilizat si pentru clustering si ne ofera lucrul cu multi algoritmi de data mining.

**Java Text Categorizing Library**

<http://textcat.sourceforge.net/>

Java Text Categorizing Library ( JTCL ) este o implementare a librăriei libTextCat ce a fost inițial concepută pentru " ghicirea " limbii . Poate fi folosit în scopul de a pune categorii texte în topicuri arbitrar alese calculând aproprierea fingerprints ce reprezintă categoriile . Algoritmul de analizat textul e bazat pe tehnica de clasificare descrisă în Cavnar & Trenkle, "N-Gram-Based Text Categorization".

Această librărie a fost scrisă de Knallgrau New Media Solutions și este folosită pe tagthe.net , un serviciu web ce extrage meta informațiile de la o anumită resursă dată . Java Text Categorizing Library este utilizată să determine limba în care e scrisă resursa dată și e planificată să fie folosită pentru a determina topicuri în viitorul recent . Ultima versiune apărută este 1.0

Limbi suportate : albaneză , danemarca , olandeza , engleza , finlandeza , franceza , germana , maghiara , italiana , norvegiana , poloneza , slovaca , slovena , spaniola , suedeza

Clasa TextCategorizer poate lua de exemplu un argument dat la linia de comandă și-l trimite mai departe spre metoda de ce pune pe categorii instanțiată de LanguageGuesser ce returnează limba recunoscută drept un obiect de tip String .

Fișierele FingerPrints pot fi create programatic executând jar-ul java adăugând la comandă opțiunea

"-createfp" .

Un exemplu de utilizare ar fi următorul : java -jar textcat.jar -createfp school.txt

Executând această comandă , textcat.jar va căuta după un fișier numit "school.txt" în execuția căii sale și îl va folosi în contextul creării a unui nou fișier fingerprint numit "school.lm" în executarea căii .

**Mallet**

<http://mallet.cs.umass.edu/index.php>

MALLET este un pachet bazat pe limbajul de programare Java, printre ale cărui funcționalități se regăsesc următoarele, care ne-ar servi realizării proiectului:

* Clasificarea documentelor;
* Clusterizare;
* Extragere de informații din text;

MALLET include instrumente sofisticate de clasificare a documentelor: modalități

eficiente de transformare ale textelor în ”caracteristici” (împărțirea acestora pe categorii), o varietate largă de algoritmi incluzând Naïve Bayes (pentru clasificare), Maximum Entropy, DecisionTrees (arbori de decizie) și cod util pentru evaluarea randamentului / performanței clasificatorului utilizat.

În afară de clasificare, MALLET include utilitare pentru ”sequencetagging” (reprezintă un tip de pattern matching), implementate cu algoritmi precum ”Hidden Markov Models”, ”Maximum Entropy Markov Models” și ”ConditionalRandomFields”.

Pachetul MALLET dispune de un set de instrumente pentru ”topic modelling”, manieră utilă pentru analiza colecțiilor largi de texte neetichetate. Acest set de instrumente implementări eficiente, bazate pe șabloane, ale ”Latent Dirichlet Allocation”, ”PachinkoAllocation” și ”Hierarchical LDA”.

Întregul pachet este Open Source Software, sub clauzele ”Common Public License”.

-Dictionar <http://www.tutorialspoint.com/java/java_dictionary_class.htm>

**-JAXP**

Formatul in care vor fi salvate atat textele cat si categoriile va fi XML. Vom pastra adnotarile pentru fiecare text. O biblioteca utila pentru lucrul cu XML-uri in Java este JAXP : <https://docs.oracle.com/javase/tutorial/jaxp/> API-ul pe care o sa il utilizam va fi cel pentru DOM deoarece ne permite sa manipulam datele usor si este potrivit pentru aplicatiile interactive si care prelucreaza datele constant.

**Entity Annotation**

<https://gate.ac.uk/sale/tao/splitch21.html#x26-50800021.3>

**Noun phrase classification**

<https://gate.ac.uk/sale/tao/splitch23.html#x28-52800023.2>