# Extragerea trăsăturilor

## 1 Concepte de bază

Text mining se referă la procesul de minerit a documentelor în format text. Documentele text provin din diferite surse, cum ar fi: articole de ştiri, lucrări de cercetare, cărţi, biblioteci digitale, mesaje e-mail, pagini web şi fluxuri RSS etc. Aceste tipuri de date sunt considerate a fi semi-structurate deoarece nu au o structură complet deterministă – ca la bazele de date. Există tehnici de regăsire a informaţiilor, cum ar fi metodele de indexare a textului, dezvoltate pentru a manevra astfel de documente nestructurate. În general, metodele de regăsire a informaţiilor (metodele de text mining) se împart în două categorii: metode care văd regăsirea de informaţie ca o **problemă de selecţie** a unor documente şi metode care oferă o **ierarhie de documente**. Această ierarhie se bazează pe un scor obţinut de fiecare document în funcţie de cât de similare sunt acestea faţă de ceea ce doreşte utilizatorul (interogarea formulată).
Cea mai des utilizată abordare pentru reprezentarea documentului este utilizarea modelului spaţiului vectorial (Vector Space Model - VSM). Ideea de bază a modelului VSM este următoarea: reprezentăm un document, ca vector într-un spaţiu *n*-dimensional corespunzător pentru toate cuvintele cheie conţinute în setul de documente şi vom folosi o măsură de similaritate specifică pentru a calcula similaritatea dintre vectorul de interogare şi vectorul documentului. Valorile similarităţii pot fi apoi utilizate pentru ierarhizarea documentelor. Pornind de la un set $D$ de documente şi $t$ termeni (cuvinte), putem modela fiecare document ca fiind un vector $v$ într-un spaţiu $n$ dimensional ortogonal $R^n$. Coordonata $j$ a lui $v$ este un număr (frecvenţă de termeni – care se poate şi pondera) care măsoară importanţa termenului $j$ pentru documentul dat: este de obicei 0 dacă documentul nu conţine termenul şi diferit de zero în rest. Presupunând acum că la documente similare se aşteaptă să aibă o frecvenţă similară a termenilor putem măsura similaritatea prin compararea frecvenţelor cuvintelor de bază.

## 2 Baza de date Reuters

Colecţia de date text Reuters-2000 conţine 806.791 ştiri publicate de agenţia Reuters între anii 1996-1997. Articolele sunt preclasificate de către Reuters în trei categorii mari astfel: în funcţie de ramura industrială la care face referire articolul existând 870 categorii, în funcţie de zona geografică unde există 366 categorii distincte şi după o categorie (topic) propuse de Reuters, în funcţie de conţinutul ştirii astfel existând 126 de categorii diferite.

Fiecare articol este salvată într-un fişier „XML” în care informaţiile sunt grupate în noduri. Structura unui astfel de fişier xml este:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
-<newsitem xml:lang="en" date="1997-03-29" id="root" itemid="475078">
      <title> </title>
      <headline> </headline>
      <byline> </byline>
      <dateline> </dateline>
      +<text>
      <copyright>(c) Reuters Limited 1997</copyright>
      +<metadata>
 </newsitem>
```

Cele mai importante noduri sunt:

- nodul „<title>” care conţine titlul articolului,
- nodul „<text>” care conţine textul propriu-zis al articolului. În cadrul acestui nod fiecare paragraf din articol este delimitat de tagurile „ ” şi „ ”
- nodul „<metadata>” care conţine categoriile în care este grupat articolul. În cadrul acestui nod există subnoduri de tipul „codes class” care grupează articolul respectiv în funcţie de cele trei categorii. Pentru categoria „regiune” numele clasei subnodului este „<codes class="bip:countries:1.0">…</code>”, pentru categoria referitoare la ramura industrială nodul este „<codes class="bip: industry:1.0">…</code>” iar pentru categoria propusă de către Reuters pentru acel articol numele nodului este „<codes class="bip: topics:1.0">…</code>”. În nodul „<metadata>” mai sunt câteva sub-noduri de tip valoare care conţin informaţii despre data apariţiei acelui articol, titlul, agenţia care la emis, limba, locul apariţiei şi altele. Informaţiile incluse în aceste noduri sunt prescurtări (coduri) pentru fiecare categorie şi care sunt detaliate în fişiere separate. Fiecare articol poate fi inclus în una sau mai multe categorii.

Fişierele extrase din această bază de date au fost salvate în două foldere. Fişierele din folderul „Training” vor fi folosite în procesul de text mining în etapa de antrenare a algoritmilor de învăţare, iar fişierele din folderul „Testing” vor fi folosite în procesul de text mining în etapa de testare şi evaluare a rezultatelor învăţării. Mai multe informaţii despre codificarea fişierelor Reuters se poate găsi la adresa codificare Reuters .

# 3 Etapa de extragere a trăsăturilor

În această etapă documentele din format text sunt transpuse într-o reprezentare care să permită utilizarea uşoară în procesul de învăţare. Reprezentarea aleasă este cea de vectori de frecvenţe de cuvinte. Astfel fiecare document este reprezentat printr-un vector în care fiecare element conţine cuvântul şi numărul de

apariţii ale acestuia în documentul respectiv. În această etapă se parcurge fiecare document pe rând şi se extrag cuvintele, eliminându-se semnele de punctuaţie, cifrele şi oricare alte tag-uri care nu rprezintă text propriu-zis. Tot în această etapă se vor elimina cuvintele de legătură şi se va extrage rădăcina cuvântului.

## 3.1 Stopwords

Documentele text conţin o serie de cuvinte care nu transmit nici o informaţie utilă dar care sunt folosite în limbajul curent pentru a lega între ele cuvintele din propoziţie. Aceste cuvinte sunt numite cuvinte de legătură (stopwords) şi nu vor fi luate în considerare în procesul de text mining. De exemplu câteva cuvinte de legătură pentru limba engleză ar fi „and, or, at, be …”. Pentru cuvintele de legătură în limba engleza folosiţi fişierul „stopwords.txt”

## 3.2 Stemming

O altă etapă în procesul de extragere a trăsăturilor este acea de extragere a rădăcinii cuvântului. De obicei un cuvânt apare în mai multe forme în text, de exemplu cuvântul „child” şi cuvântul „children” şi care înseamnă acelaşi lucru. De aceea în procesul de extragere a cuvintelor (trăsăturilor) există o etapă de extragere a rădăcinii cuvintelor iar în reprezentarea documentelor se vor folosi doar rădăcinile cuvintelor. Pentru extragerea rădăcinii cuvintelor se poate folosi clasa „Porter” sau clasa „Lovin”.

## 3.3 Vector space model

În reprezentarea VSM fiecare document este reprezentat ca un vector în care pe fiecare poziţie se găseşte frecvenţa de apariţie a unui cuvânt în acel document. Pentru fiecare document în parte se va memora câte un astfel de vector care conţine cuvântul şi un contor pentru numărul de apariţii ale acelui cuvânt în documentul curent. În paralel se va memora şi un vector global care conţine toate cuvintele care apar în toate documentele. Altfel se parcurge pe rând câte un document şi se extrage cuvintele din acele documente. Fiecare cuvânt este introdus în vector global doar dacă acesta nu există deja. Cuvântul extras este introdus în vectorul local al documentului curent dacă acesta nu există sau se incrementează contorul corespunzător, dacă acesta există. După ce s-au procesat toate documentele vectorul global va conţine toate cuvintele care apar în toate documentele (acesta mai este denumit şi dicţionarul setului de date curent). Acest vector caracterizează întreg setul de documente iar dimensiunea lui reprezintă dimensiunea spaţiului de reprezentare a tuturor documentelor din acel set.

Fiecare vector iniţial creat pentru un document este apoi modificat astfel încât să devină de lungimea vectorului global, specificându-se valoarea 0 pe poziţiile cuvintelor ce nu apar în documentul respectiv, pe celelalte poziţii specificându-se frecvenţa de apariţie a cuvântului. Fiecare vector de document va respecta ordinea cuvintelor din vectorul global. Astfel la sfârşit fiecare vector al documentului va avea

aceeaşi dimensiune (indiferent de dimensiunea iniţială a documentului) şi va conţine pe aceeaşi poziţie valorile aceluiaşi cuvânt.

Deoarece memoria necesară pentru a stoca aceşti vectori este destul de mare o variantă mult mai simplă este de a memora doar valorile diferite de 0. Soluţia propusă este cea folosită în cazul reprezentării vectorilor rari în care se memorează prima dată poziţia din vector şi apoi valoarea corespunzătoare. Acest mod de memorare este eficient în acest context deoarece fiecare vector conţine foarte multe valori de 0.

Exemplu:

Considerăm că avem 3 documente:

*D1: Classification and clustering are learning algorithms.*

*D2: Classification is a supervised learning.*

*D3: Clustering is an unsupervised learning.*

Vectori de reprezentare pentru fiecare document sunt

D1:

| classification:1 | clustering:1 | learning:1 | algorithm:1 |
|---|---|---|---|

D2:

| classification:1 | superviseed:1 | learning:1 |
|---|---|---|

D3:

| clustering:1 | unsupervised:1 | learning:1 |
|---|---|---|

Vectorul global pentru toate 3 documentele este:

| algorithm | classification | clustering | learning | supervised | unsupervised |
|---|---|---|---|---|---|

Fiecare vector de document va deveni de lungime 6 şi va memora doar frecventa de apariţie a cuvintelor din acel document astfel:

| Document | algorithm | classification | clustering | learning | supervised | unsupervised |
|---|---|---|---|---|---|---|
| D1 | 1 | 1 | 1 | 1 | 0 | 0 |
| D2 | 0 | 1 | 0 | 1 | 1 | 0 |
| D3 | 0 | 0 | 1 | 1 | 0 | 1 |

Reprezentarea ca şi vectori rari a celor 3 documente este (valoarea de dinaintea simbolului „:” reprezintă poziţia cuvântului iar valoarea de după simbolul „:” reprezintă frecvenţa de apariţie a cuvântului):

D1 1:1 2:1 3:1 4:1

D2 2:1 4:1 5:1

D3 3:1 4:1 6:1

**Problemă:** Să se creeze şi să se salveze într-un fişier vectorii de frecvenţe pentru articolele date. Pentru fiecare articol se vor extrage informaţiile din nodurile „<title>”, „<text>” şi „<codes class="bip:topics:1.0">”. Informaţiile extrase din nodurile „<title>” şi „<text>” vor fi considerate ca şi conţinutul articolului şi vor fi folosite în etapa de formare a vectorilor de frecvenţe de cuvinte.

Informaţiile extrase din nodul „<codes>” vor fi considerate categoriile din care face parte articolul respectiv. Fiecare articol poate face parte dintr-o categorie sau mai multe categorii. Pentru fiecare categorie se va păstra doar codul categoriei. În fişierul rezultat se va salva fiecare articol pe câte o linie care conţine articolul în format de vector rar urmat de simbolul „#” apoi cu spaţiu codurile topicurilor pentru acel articol şi după următorul simbol „#” folderul de unde a fost luat articolul respectiv (cuvântul Traning sau Testing).

Exemplu. Cele 3 documente din exemplul de mai sus pot fi salvate în fişier astfel (considerăm că „ml”, „cla” şi „clu” sunt topicurile celor trei documente):

@data

1:1 2:1 3:1 4:1 # ml cla clu # training

2:1 4:1 5:1 # ml cla # training

3:1 4:1 6:1 # ml clu # testing