Treball Final de Grau

NewsAnalyzer: un analitzador de notícies etiquetades

Alumne: Rafel Castaño Ribes

Consultor: David Isern Alarcón

Estudis: Enginyeria informàtica

Itinerari: Computació

Àrea: Intel·ligència artificial

Data: 15 de juny de 2011

A la Mòni, la meva dona,
per la seva paciència i suport,
i a l'Aleix i la Duna,
a tots tres, amb tot el meu amor.

Index

1 Resum 1
2 Introducció 2
2.1 Justificació i motivació del TFG 2
2.2 Objectius 3
2.3 Estructuració del treball 4
2.4 Planificació i execució del treball 5
2.4.1 Relació de tasques 5
2.4.2 Planificació temporal 6
2.5 Desviacions respecte a la planificació inicial 7
3 Estat de l'art i punt de partida del TFG 8
3.1 Fonts de notícies i les seves APIs 8
3.2 Treballs similars a aquest TFG 10
3.3 Punt de partida del TFG 13
3.3.1 Quant a la recol·lecció de notícies 13
3.3.2 Quant al procediment de normalització 13
3.3.3 Quant al procediment de classificació de les etiquetes 13
3.3.4 Quant a l'avaluació del model 14
4 Fase de recollida de requisits 15
4.1 Especificació dels requisits 15
4.1.1 Requisits funcionals 15
4.1.2 Requisits no funcionals 15
4.2 Identificació dels casos d'ús 16
4.2.1 Descripció textual dels casos d'ús 17
5 Fase d'anàlisi 19
5.1 Model de domini 19
5.1.1 Diagrama UML estàtic de classes 19
5.2 Identificació de les classes frontera i de control i de les operacions. 21
5.2.1 Diagrama de col·laboració del cas d'ús: iniciar el recol·lector 21
5.2.2 Diagrama de col·laboració del cas d'ús: iniciar l'analitzador 21
5.2.3 Diagrama de col·laboració del cas d'ús: visualitzar resultats per font 22
5.2.4 Diagrama de col·laboració del cas d'ús: visualitzar resultats per etiqueta 22
5.3 Modelització de la Interfície Gràfica d'Usuari 23
5.3.1 Pantalla Menu 23
5.3.2 ArticleRetrieverGUI: 24
5.3.3 AnalyzerGUI: 24
5.3.4 ResultsPerSourceGUI: 25
5.3.5 ResultsPerTagGUI: 25
5.4 Relació de classes de control identificades 26
5.4.1 Gestor Menú: 26
5.4.2 ArticleRetreiverModule 26
5.4.3 AnalyzerModule 26
5.4.4 ResultsExplorer 26
6 Fases de disseny i d'implementació 27
6.1 Elecció del llenguatge de programació: Java 27
6.2 Elecció del sistema gestor de base de dades: MySQL 27
6.3 Elecció de la versió de WordNet 27
6.4 Llibreries externes 28
6.4.1 JDBC de MySQL 28

6.4.2 JSON simple 28
6.4.3 Java API for WordNet Searching (JAWS) 28
6.5 Disseny de la persistència 29
6.5.1 Model Relacional 29
6.5.2 Gestió de la persistència (package *dataModel.DAOs*) 31
6.6 Disseny del mòdul recol·lector de notícies (package *retrievers*) 33
6.6.1 Recol·lectors 35
6.6.2 The New York Times (subpackage *nyt*) 36
6.6.3 The Guardian (subpackage guardian) 37
6.6.4 Interfície gràfica d'usuari (subpackage gui) 39
6.6.5 Aspectes que han quedat pendents 39
6.7 Mòdul analitzador (package analyzer) 39
6.7.1 Classe AnalyzerModule 40
6.7.2 Submòdul d'utilitats (subpackage *utils*) 41
6.7.3 Classe FacetPredictor 42
6.7.4 Classe StringComparator 43
6.7.5 Accés a recursos (subpackage *frontEnds*) 44
6.7.6 Accés a la Wikipedia (subpackage *frontEnds.wikipedia*) 44
6.7.7 Accés a WordNet (subpackage *frontEnds.wordnet*) 45
6.7.8 Accés a DBpedia (subpackage *dbpediaFrontEnd*) 47
6.7.9 Interfície Gràfica d'Usuari (subpackage *gui*) 49
6.7.10 Aspectes que han quedat pendents 49
6.8 Mòdul Visualitzador de resultats (package *resultsExplorer*) 50
6.8.1 Classe ResultsExplorerModule 50
6.8.2 Vistes dels resultats (subpackage *gui*) 51
6.8.3 Representació de les dades (subpackage *gui.tables*) 51
6.8.4 Vista d'avaluació orientativa dels resultats obtinguts 53
6.8.5 Aspectes que han quedat pendents 53
6.9 Pantalla principal i menú de l'aplicació (package *gui*) 53
6.10 Lliurable final 53
7 Proves i discussió de resultats 55
7.1 Proves en el procediment de recol·lecció de notícies 55
7.1.1 Volum de les dades 55
7.1.2 El factor temps en la recol·lecció 56
7.1.3 Discussió dels resultats 56
7.2 Proves en el procediment d'anàlisi de les notícies 57
7.2.1 Model de referència de Borort, S. (2009) 57
7.2.2 Millores introduïdes en aquest treball 58
7.2.3 Discussió dels resultats 60
7.3 Proves quant al compliment dels objectius d'aquest treball 61
7.3.1 Temàtiques globals (entre totes les fonts) 61
7.3.2 Temàtiques més habituals al diari The Guardian 62
7.3.3 Temàtiques més habituals al diari The New York Times 63
7.3.4 Temàtiques més en un període de temps concret 63
7.3.5 Discussió dels resultats 64
8 Conclusions i propostes de millora 65
8.1 Conclusions 65
8.2 Propostes de millora 66
9 Bibliografia 67
Annexos 69

Annex A – Mitjans emprats 69
Annex B – Manual de compilació del programa 70
Annex C – Manual d'instal·lació de l'aplicació 72
Annex D – Manual d'usuari 74
1. Execució 74
2. Relació de menús i les seves funcionalitats 74
3. Mode debug 80
Annex E – Índex de figures 81

1 Resum

En aquest treball es pretén abordar la problemàtica de confirmar o refutar si determinades fonts de notícies online mostren algun tipus de biaix que a priori un lector podria detectar per simple intuïció.

Per simplificar les tasques d'anàlisi es treballa amb fonts de notícies que disposen d'APIs d'accés als seus articles i que, a més a més, proporcionen anotacions semàntiques (etiquetes) associades a cada notícia a mode de classificació d'aquestes.

Per complir amb els objectius plantejats s'analitza i millora un mètode descrit en la bibliografia que permet dur a terme una anàlisi de les etiquetes per tal d'obtenir i aplicar un vocabulari comú a les diferents fonts (procediment de normalització).

El programari resultant es presenta com una aplicació implementada en Java i MySQL que recol·lecta notícies anotades semànticament de diferents fonts de notícies online (els diaris The Guardian i The New York Times), les analitza i permet visualitzar els resultats en funció del vocabulari normalitzat per tal d'extreure conclusions sobre quins són els temes més tractats per cada font.

Finalment, s'analitzen els resultats obtinguts, es discuteixen i s'extreuen una sèrie de conclusions sobre el mètode de normalització i classificació emprats i es proposen possibles millores per al futur de l'aplicació.

2 Introducció

Es considera que actualment vivim en la societat de la informació i la comunicació. Això vol dir que ens trobem sota un bombardeig constant d'informació que prové de fonts molt diverses i en un gran volum. Aquesta quantitat tan immensa de dades fa que sigui impossible el seu processament i assimilació totals per part de la ment humana.

Sota aquest context han sorgit una gran quantitat d'iniciatives que recol·lecten informació i extreuen coneixement a través de la xarxa per tal de facilitar la tasca humana d'anàlisi de la informació.

Aquest és el context sota el qual es presenta aquest Treball Final de Carrera.

2.1 Justificació i motivació del TFG

Es diu que en determinades fonts de notícies la secció relativa a esports o a un equip en concret té molt més gruix que altres seccions que potser són tant o més importants. Aquestes afirmacions només se sustenten en la percepció que els lectors poden tenir sobre una font de notícies i són realment subjectives.

Una forma de confirmar o rebatre aquests tipus d'afirmacions és realitzar un estudi de les dades que un diari publica al llarg d'un temps. En Intel·ligència Artificial existeixen tècniques de Mineria de Dades per quantificar cadascuna d'aquestes noticies i fer-ne un estudi quantitatiu.

Actualment, la Web 2.0 dóna diferents facilitats per la recollida de les noticies. Per exemple, els diaris The New York Times (Estats Units) i The Guardian (Regne Unit) tenen una API que permet recollir noticies que estan anotades semànticament en períodes concrets de temps.

API del diari The New York Times:
http://developer.nytimes.com/docs/read/article_search_api

API del diari The Guardian:
http://www.guardian.co.uk/open-platform

Aquestes API poden servir per composar un sistema d'aprenentatge i extreure'n conclusions. En aquest cas, les conclusions poden ser de l'estil **"el beisbol ocupa un 70% de les noticies"**, o **"el 80% de les noticies estan relacionades amb l'Obama"**.

Les conclusions d'aquest estudi poden resultar interessants per a tercers en el sentit que si gràcies a aquest estudi es conclou que, per exemple, les noticies relacionades amb la política representen un 90% del total d'un diari concret, es podrà intuir que no es tracten altres temes que potser interessen a aquests tercers.

L'aplicació resultant serà una aplicació local que interaccionarà amb les diferents APIs i estarà implementada en Java, tot emmagatzemant la informació obtinguda i generada en una base de dades MySQL.

2.2 Objectius

Els objectius mínims que es pretenen assolir en aquest TFG són els següents:

a) Oferir un mecanisme automatitzat de recol·lecció de notícies anotades semànticament (etiquetades) de diferents fonts on-line (com a mínim dues fonts). Cal que aquest procés sigui escalable, que la recollida pugui ser diària o setmanal, i que no tingui límits de volum.

b) Donat que cada font pot emprar un vocabulari diferent per a etiquetar les seves notícies, oferir un mecanisme que permeti unificar els diferents vocabularis per tal d'emprar una semàntica única en l'etiquetatge de les notícies.

c) Per a aquelles paraules que puguin tenir un significat ambigu, definir algun algorisme que permeti determinar a quina de les diferents accepcions fa referència el terme.

d) Categoritzar les etiquetes tot identificant el tipus d'ens a què es refereixen cadascuna d'elles.

e) Obtenir una representació lògica de les dades que permeti emmagatzemar les notícies, les etiquetes i els resultats de l'anàlisi en una mateixa base de dades independentment de la font de la que procedeixin les notícies.

f) Oferir un entorn gràfic que permeti analitzar els resultats dels procediments anteriors així com visualitzar ràpidament quins són els temes majoritaris tractats en cada font d'informació.

g) Oferir el producte com una aplicació local que integri totes les funcionalitats anteriors.

h) Oferir el producte com a el conjunt de fitxers que integren el codi font i una sèrie de scripts que permetin compilar el programa així com generar un directori que pugui ser el distribuïble, juntament amb els scripts responsables de l'execució de l'aplicació.

i) Oferir els manuals adients per tal de dotar a l'usuari final de les instruccions per a compilar, executar i utilitzar el programa.

j) Documentar tot el projecte adientment.

Altres objectius assolibles són:

k) Dur a terme algun procediment de descoberta de coneixements sobre les dades.

2.3 Estructuració del treball

El treball es pot desglossar en quatre grans blocs:

1. Recollida, tractament inicial i emmagatzematge de les notícies

   Assoliment dels objectius a) i part del e)

2. Anàlisi, categorització i classificació de les etiquetes

   Assoliment dels objectius b), c), d) i e)

3. Presentació dels resultats

   Assoliment dels objectius f), g), h) i i)

4. Documentació del projecte

   Assoliment de l'objectiu j)

Els tres primers blocs es poden desenvolupar de forma seqüencial i progressiva, mentre que el quart bloc es pot desenvolupar de forma transversal al llarg de tot el projecte.

L'objectiu k), de descoberta de la informació, s'ha preferit eliminar-lo de l'estructuració del treball degut a què finalment no hi ha hagut temps per desenvolupar-lo.

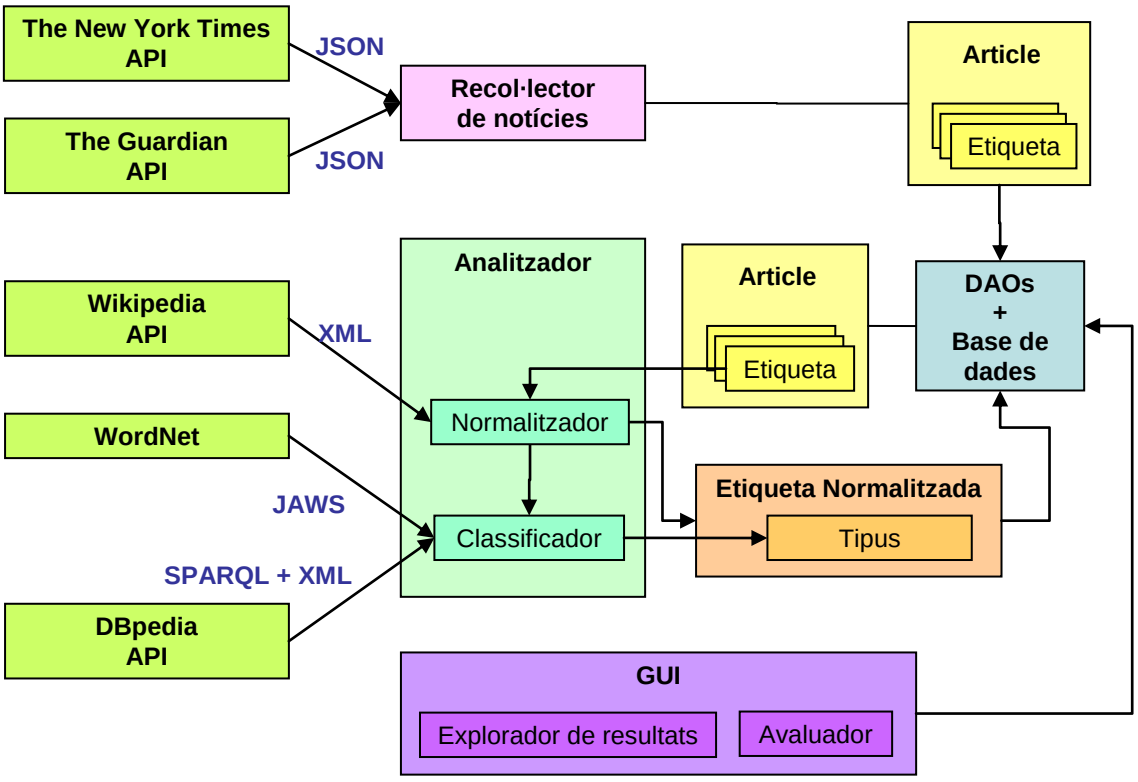


Figura 1 – Estructuració del treball

2.4 Planificació i execució del treball

Tot seguit es presenta la relació de tasques en què s'ha desglossat el projecte així com l'estimació inicial del temps necessari per realitzar-les, la consecució d'aquestes i la desviació respecte al temps programat.

2.4.1 Relació de tasques

La relació de tasques en què s'ha desglossat aquest TFG és la següent:

Bloc 1: Recollida, tractament inicial i emmagatzematge de les notícies

1. Recollida de requisits
   1.1. Recerca sobre les diferents APIs d'accés a notícies etiquetades i el seu funcionament.
2. Anàlisi
   2.1. Obtenció i descripció dels casos d'ús
   2.2. Obtenció del diagrama de classes estàtic
3. Disseny
   3.1. Disseny de la persistència
   3.2. Disseny de l'aplicació
4. Implementació
5. Documentació

Bloc 2: Anàlisi, categorització i classificació de les etiquetes

1. Recollida de requisits
   1.1. Recerca sobre l'estat de l'art en el procés de creació de vocabularis comuns entre diferents fonts.
   1.2. Recerca sobre l'estat de l'art en la categorització i classificació d'etiquetes.
2. Anàlisi
   2.1. Determinar el procediment de normalització del vocabulari d'etiquetatge.
   2.2. Determinar les categories o clústers sota els quals es classificaran les diferents etiquetes de les notícies.
   2.3. Determinar el procediment mitjançant el qual es decidirà a quina categoria pertany cada etiqueta (classificació).
   2.4. Determinar el procediment d'avaluació dels mètodes anteriors.
3. Disseny
   3.1. Disseny del procediment de normalització de les etiquetes.
   3.2. Disseny del procediment de classificació de les etiquetes.
   3.3. Disseny del procediment d'avaluació.
4. Implementació
   4.1. Implementació del procediment de normalització de les etiquetes.
   4.2. Implementació del procediment de classificació de les etiquetes.
   4.3. Implementació del procediment d'avaluació
5. Documentació

Bloc 3: Presentació dels resultats

1. Anàlisi
   1.1. Identificació i definició dels casos d'ús
   1.2. Identificació de les diferents pantalles i vistes que ha d'oferir l'aplicació
2. Disseny
   2.1. Prototipatge de la interfície gràfica d'usuari (GUI)
   2.2. Disseny de la distribució de l'aplicació.
3. Implementació
4. Elaboració dels manuals de compilació, instal·lació i d'usuari.
5. Documentació

Bloc 4: Integració i compleció de la documentació

1. Elaboració de la memòria
2. Elaboració de la presentació

2.4.2 Planificació temporal

Quant a la planificació temporal del TFG cal tenir present que el temps és molt limitat i que hi ha quatre dates que són clau:

- 19 Març 2011 : Lliurament de la PAC1
- 16 Abril 2011 : Lliurament de la PAC2
- 21 Maig 2011 : Lliurament de la PAC3
- 15 Juny 2011 : Lliurament final

Partint d'aquesta base, la planificació inicial va ser la que es resumeix en la següent taula i diagrama de Gantt:

WBS	Nom	Comença	Acaba	Feina
1	**Bloc 1: Recol·lector de notícies**	**17 març**	**2 abr**	**34d**
1.1	**Lògica de l'aplicació**	**17 març**	**26 març**	**20d**
1.1.1	Desenvolupament	17 març	26 març	10d
1.1.2	Documentació	17 març	26 març	10d
1.2	**Persistència**	**27 març**	**2 abr**	**14d**
1.2.1	Desenvolupament	27 març	2 abr	7d
1.2.2	Documentació	27 març	2 abr	7d
2	**Bloc 2: Analitzador d'etiquetes**	**3 abr**	**21 maig**	**84d**
2.1	**Normalització**	**3 abr**	**16 abr**	**28d**
2.1.1	Desenvolupament	3 abr	16 abr	14d
2.1.2	Documentació	3 abr	16 abr	14d
2.2	**Classificació**	**17 abr**	**30 abr**	**28d**
2.2.1	Desenvolupament	17 abr	30 abr	14d
2.2.2	Documentació	17 abr	30 abr	14d
2.3	**Avaluació dels procediments**	**1 maig**	**7 maig**	**14d**
2.3.1	Desenvolupament	1 maig	7 maig	7d
2.3.2	Documentació	1 maig	7 maig	7d
2.4	Descoberta de coneixement	8 maig	21 maig	14d

WBS	Nom	Comença	Acaba	Feina
3	**Bloc 3: Presentació dels resultats**	**22 maig**	**30 maig**	**18d**
3.1	**Interfície gràfica d'usuari**	**22 maig**	**28 maig**	**14d**
3.1.1	Desenvolupament	22 maig	28 maig	7d
3.1.2	Documentació	22 maig	28 maig	7d
3.2	**Lliurable**	**29 maig**	**29 maig**	**2d**
3.2.1	Desenvolupament	29 maig	29 maig	1d
3.2.2	Documentació	29 maig	29 maig	1d
3.3	Manuals de compilació i d'usuari	29 maig	30 maig	2d
4	**Documentació final del projecte**	**29 maig**	**11 juny**	**14d**
4.1	Memòria	29 maig	6 juny	9d
4.2	Presentació digital	7 juny	11 juny	5d

Figura 2 – Taula resum de les tasques del TFG

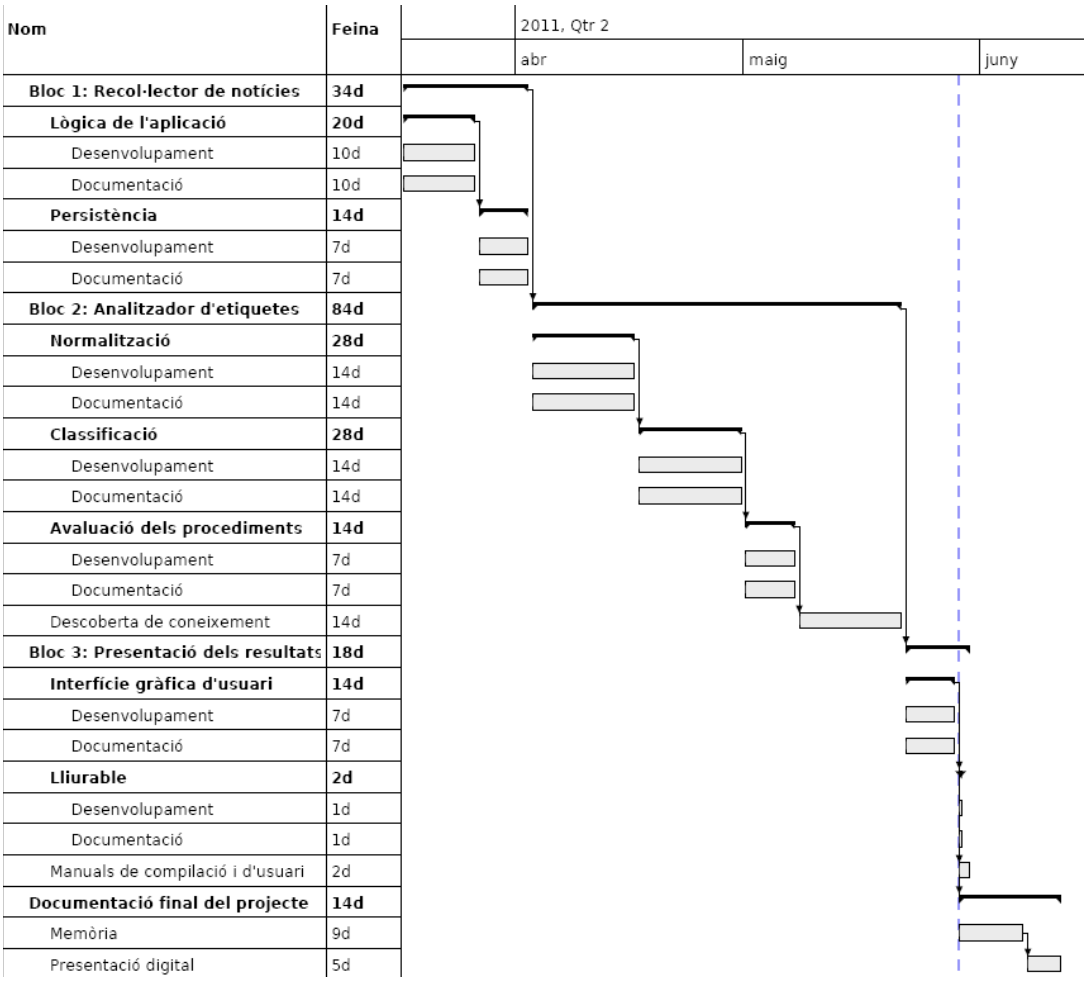


Figura 3 – Diagrama de Gantt amb la planificació del TFG

2.5 Desviacions respecte a la planificació inicial

En la planificació inicial s'incloïa la tasca de desenvolupament de la funcionalitat de descoberta de coneixement. Per diverses circumstàncies no hi ha hagut temps de desenvolupar-la i el seu temps ha estat ocupat per tot el bloc segon i part del primer, que s'han allargat considerablement.

3 Estat de l'art i punt de partida del TFG

Abans d'entrar a parlar de la part tècnica del TFG convé aturar-se un instant i fer una ullada a la situació actual de la tecnologia en l'àmbit en què se centra aquest projecte.

Per una banda cal constatar que efectivament hi ha diferents fonts de notícies que permeten la descàrrega programàtica de les seves notícies, o com a mínim dels seus titulars, juntament amb una classificació d'aquestes mitjançant anotacions semàntiques o **etiquetes** (**tags** en anglès).

D'altra banda, cal fer una mica de recerca sobre quins han estat els esforços que s'han dut a terme en el camp de la integració de sistemes etiquetats per tal que aquests puguin ésser comparables, és a dir, quines han estat les iniciatives en el camp de la normalització d'etiquetes.

Un cop obtinguts els resultats de la recerca per a aquests dos aspectes del treball convé analitzar-los amb detall per tal d'elaborar una possible estratègia inicial per abordar els diferents objectius que es plantegen en aquest projecte.

Tot seguit es resumeixen els resultats de la recerca i de l'anàlisi duts a terme.

3.1 Fonts de notícies i les seves APIs

Estudis recents afirmen que en els estats units les fonts de notícies online han superat en audiència als diaris convencionals (Pew Research Center for the People & the Press, 2011, The State of the News Media 2011).

La forta explosió d'aquests mitjans a Internet, ha fet que comencin a explorar noves vies d'expansió per tal de guanyar força a la xarxa. Entre aquestes vies es troba el desenvolupament d'interfícies de programació per a aplicacions (APIs) de tipus obert per tal d'oferir als programadors d'aplicacions d'arreu del món accés a les seves notícies, o als titulars i resums d'aquestes, a través d'Internet.

El fet que aquestes APIs siguin de tipus obert és fonamental, ja que poc a poc i de forma espontània, donat a què la premsa online va guanyant pes, van sorgint noves aplicacions que les utilitzen i per tant, aquests productors de notícies van guanyant presència en el mercat.

Entre les diferents fonts de notícies que ofereixen aquests tipus d'APIs podem citar les següents:

- The New York Times (NYT) (Estats Units)
  http://developer.nytimes.com/docs/read/article_search_api

- The Guardian (Regne Unit)
  http://www.guardian.co.uk/open-platform

- British Broadcasting Corporation (BBC) (Regne Unit)

  http://backstage.bbc.co.uk/data/Data
  http://ideas.welcomebackstage.com/data

- National Public Radio (NPR) (Estats Units)

  http://www.npr.org/api/index

Entre les diferents fonts de notícies mencionades s'ha decidit iniciar el treball amb The New York Times i The Guardian per dos motius fonamentals:

1) Proporcionen anotacions semàntiques dels seus continguts que curosament han atorgat les persones encarregades de catalogar cada notícia, sovint partint de la base d'un ampli thesaurus de l'empresa.

2) Ofereixen una API més completa.

A nivell tècnic convé destacar el següent:

- Mètode de consulta:

  Ambdues APIs permeten l'accés als continguts mitjançant una sol·licitud HTTP que utilitza el mètode GET. En aquesta sol·licitud s'hi inclouen els paràmetres que permeten delimitar quines notícies obtenir i quins camps.

  La figura 4 mostra una consulta d'exemple a The Guardian on se sol·liciten els articles publicats els dia 30 de maig de 2011, juntament amb les etiquetes (tags) assignades, tot ordenant els articles de menys a més recent i sol·licitant el resultat en format JSON.

  http://content.guardianapis.com/search?from-date=2011-05-30&order-by=oldest&format=json&show-tags=all

Figura 4 – Sol·licitud HTTP cap a l'API del diari The Guardian

- Format de la resposta:

  El format de la resposta pot variar en funció de la font que es consulta:

  - En el cas del The New York Times, es retrona en un missatge de text amb format JSON (JavaScript Object Notation), un llenguatge lleuger per a intercanviar dades.

  - En el cas del The Guardian, l'API ens permet triar entre JSON i XML (eXtensible Markup Language).

```
{
  "response": {
    "status": "ok",
    "userTier": "free",
    "total": 448,
    "startIndex": 1,
    "pageSize": 10,
    "currentPage": 1,
    "pages": 45,
    "orderBy": "oldest",
    "results": [
      {
        "id": "society/2011/may/30/social-car-failing-disabled-pensioners-says-report",
        "sectionId": "society",
        "sectionName": "Society",
        "webPublicationDate": "2011-05-30T00:01:00+01:00",
        "webTitle": "Social care failing disabled over 65s, says report",
        "webUrl": "http://www.guardian.co.uk/society/2011/may/30/social-car-failing-disabled-pensioners-says-report",
        "apiUrl": "http://content.guardianapis.com/society/2011/may/30/social-car-failing-disabled-pensioners-
says-report",
        "tags": [
          {
            "id": "society/older-people",
            "type": "keyword",
            "webTitle": "Older people",
            "webUrl": "http://www.guardian.co.uk/society/older-people",
            "apiUrl": "http://content.guardianapis.com/society/older-people",
            "sectionId": "society",
            "sectionName": "Society"
          },
          {
            "id": "society/disability",
            "type": "keyword",
            "webTitle": "Disability",
            "webUrl": "http://www.guardian.co.uk/society/disability",
            "apiUrl": "http://content.guardianapis.com/society/disability",
            "sectionId": "society",
            "sectionName": "Society"
          },
          {
            "id": "society/social-care",
            "type": "keyword",
            "webTitle": "Social care",
```


Figura 5 – Part de la resposta de l'API del The Guardian en format JSON

En la figura 5 es mostra part de la resposta de la consulta de la figura 4. En aquesta resposta es pot observar que hi ha 448 articles publicats per a aquesta data, que es distribueixen en 45 pàgines (són 10 articles per pàgina), que ens trobem en la pàgina 1, és a dir, amb els 10 primers articles publicats per a aquesta data (ja que el criteri d'ordenació és de més antic a més recent) i arribem a veure part de la descripció del primer article. De la descripció d'aquest primer article cal destacar el títol, la URL, la secció (o categoria en què classifica la font a l'article) i les etiquetes (tags).

3.2 Treballs similars a aquest TFG

De tota la bibliografia consultada convé destacar treball final de màster de Sort Borort "Semantic Classification of Social Tags for Faceted Search" (l maig de 2009) que ha estat la columna vertebral d'aquest treball. En aquest treball es descriu una plataforma basada en python que du a terme la normalització i classificació d'etiquetes de forma similar a la que es pot desitjar en aquest TFG.

Normalització

En aquest document recol·lecten informació etiquetada de diferents fonts (blogs en general) i duen a terme una normalització de les etiquetes basada en fer una cerca del mot a l'API de la Wikipedia. Si aquesta API retorna algun o alguns mots relacionats prenen el primer resultat com a bo i el fan servir com a valor per a l'etiqueta normalitzada. En el cas que no s'obtingui resposta de l'API, descarten el mot.

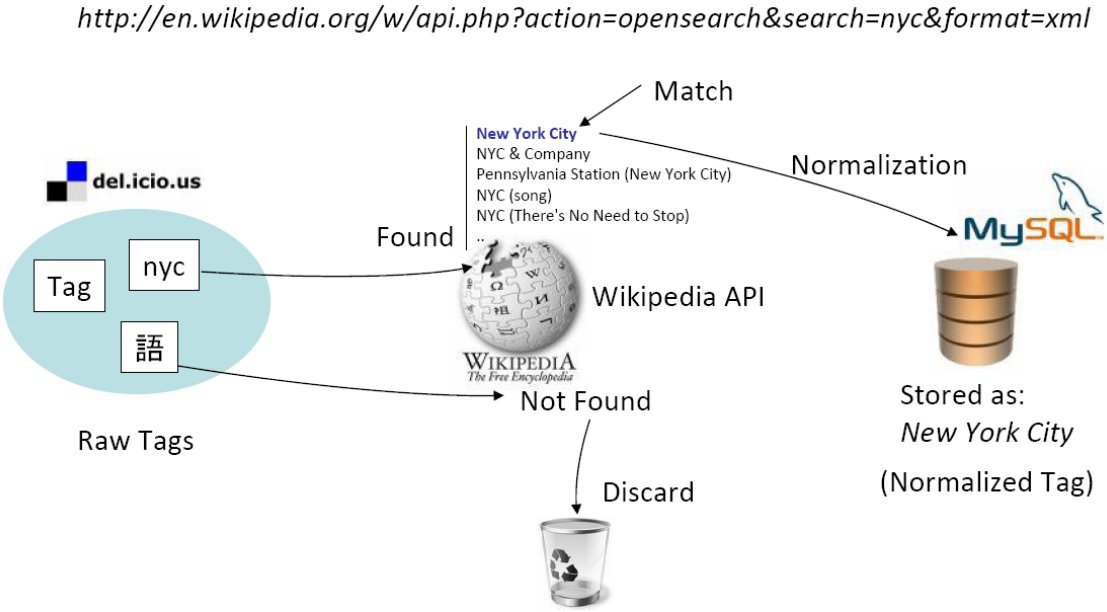


Figura 6 – Proposta de normalització d'etiquetes (Borort, S.; 2009).

Els resultats que obtenen mitjançant aquest procediment els són força favorables, ja que es poden permetre el luxe d'acceptar un cert grau d'error, ja que la finalitat de la seva aplicació és obtenir un llistat d'etiquetes candidates que mostrar a un usuari final per tal de facilitar la seva tasca d'etiquetatge.

Un altre aspecte a remarcar d'aquest procediment és que la Wikipedia és una font dinàmica que canvia constantment en el temps. Un mot pot aparèixer avui d'una determinada manera i demà d'una altra. Aquest problema però no és massa greu ja que normalment la forma principal de representar el terme no varia sovint i d'altra banda, Wikipedia està dotada d'un potent sistema de redirecció que sol reconèixer les diferents formes com sol referir-se a un terme i redirigir-nos a la seva forma principal; de fet, aquest últim aspecte és el que fa que Wikipedia els resulti una eina normalitzadora tant potent.

Un altre desavantatge de Wikipedia és que no està lliure d'ambigüitats.

D'altra banda, un gran avantatge de Wikipedia és que està força més actualitzada que qualsevol altra font més formal que requereixi de tot un procés de revisió i d'acceptació.

Classificació de les etiquetes

En el treball de Borort també es parla d'un tipus de classificació de les etiquetes molt interessant, que són els "*aspectes*" o ***facets*** en anglès. Aquest tipus de classificació normalment el que pretén és resoldre a les preguntes bàsiques de què, qui, quan, com i on. En general, el benefici d'utilitzar aquest tipus de classificadors és que permet que delimitar el rol d'una etiqueta i, per tant, facilita a l'usuari la definició de cerques més acurades.

Borort considera en el seu treball que els tipus bàsics de facets són 5:

- Persona (Qui?)
- Organització (Qui?)
- Ubicació geogràfica (On?)
- Període temporal (Quan?)
- Descriptor (Què?, com?)

Per trobar a quin facet pertany una determinada etiqueta el que fan és cercar el mot que identifica l'etiqueta a l'API de la DBpedia i observar com està classificat el terme. Si alguna de les categories en què es classifica es correspon a algun dels facets classifiquen l'etiqueta com a tal. Si en la DBpedia no troben resposta, fan una cerca a WordNet, concretament sobre la jerarquia d'hypernyms (jerarquia de categories), amb el mateix propòsit. Si no troben l'entrada en cap de les dues fonts classifiquen el terme com a descriptor.

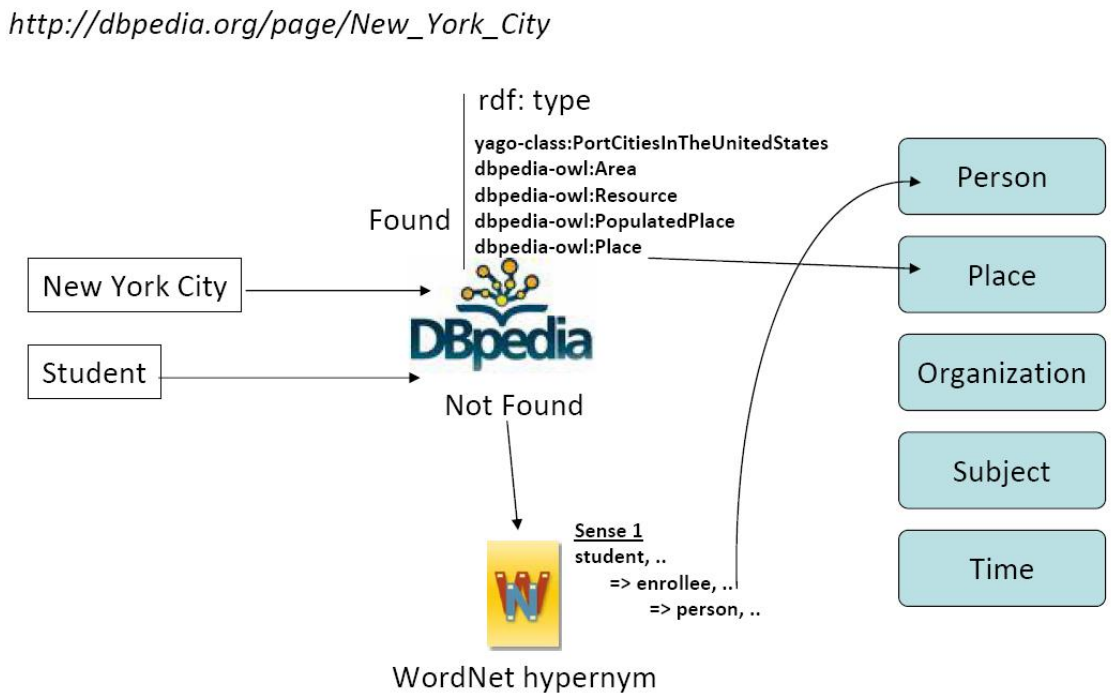


Figura 7 – Proposta de classificació d'etiquetes (Borort, S.; 2009).

DBpedia (http://dbpedia.org) és una base de dades generada a través d'un anàlisi automàtic programat sobre la Wikipedia, de la qual n'extreu coneixement. DBpedia es pot consultar online a través d'una API (http://dbpedia.org/sparql) que utilitza com a llenguatge de consulta el SPARQL (**S**PARQL **P**rotocol **a**nd **R**DF **Q**uery **L**anguage o Query Language for RDF[1]). La consulta s'invoca a través del mètode GET d'una sol·licitud HTTP i els formats de resposta són variats: XML, JSON, HTML, etc.

WordNet (http://wordnet.princeton.edu) és una iniciativa de la Princeton University que intenta definir un diccionari lliure d'ambigüitats i amb tots el termes nominals classificats jeràrquicament. Ha estat àmpliament utilitzat en diversos estudis d'anàlisis semàntics i sintàctics al llarg de la història.

[1] RDF són les sigles de Resource Description Framework, una especificació per a la descripció de recursos i conceptes publicada per la organització W3C i àmpliament utilitzada en diferents ontologies.

3.3 Punt de partida del TFG

En base a l'anàlisi que s'ha fet en l'apartat anterior (estat de l'art), l'estratègia base que s'ha decidit seguir en aquest TFG ha estat la següent:

3.3.1 Quant a la recol·lecció de notícies

Es treballarà amb l'API dels diaris The New York Times i The Guardian pels motius esmentats en el primer apartat d'aquesta secció. Es recolliran la data, títol, autors, url, etiquetes, categories i/o facets disponibles per a cada article.

3.3.2 Quant al procediment de normalització

Es mirarà si hi ha coincidència entre l'etiqueta i algun dels mots de WordNet i de Wikipedia. En cas d'èxit en alguna de les cerques se seleccionarà aquella que tingui una major similitud al mot original. Si ni WordNet ni Wikipedia són capaços d'aportar un mot de referència, l'etiqueta es descarta.

Tant el resultat de WordNet com de Wikipedia pot contenir més d'un significat per al mot de cerca. En aquest treball s'analitzarà si prendre la primera accepció com a referència tal i com fan al treball de referència és una bona opció o no i, es provaran mètodes alternatius a la recerca dels millors resultats.

3.3.3 Quant al procediment de classificació de les etiquetes

S'utilitzarà l'aproximació de classificació en facets, tot emprant els següents:

- Persona (Person)
- Organització (Organization)
- Lloc geogràfic (Geographic location)
- Període temporal o instant (Time related)
- Descriptor (Subject).

Per arribar a classificar cadascuna de les etiquetes, un cop normalitzades es cercarà el mot normalitzat a WordNet en busca de la seva jerarquia d'hypernyms (categories mare), tot cercant de quin tipus d'etiqueta es tracta. Si a WordNet no s'obté cap resultat es realitzarà el procediment anàleg a DBpedia.

En aquest cas, caldrà valorar també la possibilitat de discriminar entre les diferents accepcions que pot tenir el mot. Així doncs, Apple pot fer referència per exemple a una poma (Descriptor) o a la companyia tecnològica Apple (Organization).

3.3.4 Quant a l'avaluació del model

Donat que l'avaluació d'aquest model requeriria d'un procediment manual de revisió dels resultats de la normalització i classificació, convé proposar alguna aproximació automatitzada. En aquest sentit resulta molt interessant la font de notícies del New York Times, ja que proporciona per a moltes de les seves etiquetes els facets que els correspon. Així doncs, d'aquesta manera, una primera aproximació a la potència del model seria comparar els facets obtinguts mitjançant la classificació automàtica de les etiquetes amb l'assignació manual d'aquestes provinents dels anotadors del New York Times.

D'altra banda, aquesta aproximació no pot reemplaçar la inspecció humana dels resultats, de tal manera que l'aplicació final haurà de proporcionar una vista que permeti dur-la a terme de forma senzilla.

4 Fase de recollida de requisits

"*Els requisits expressen les necessitats i restriccions que afecten un producte de programari que contribueix a la solució d'un problema del món real i ens serveixen per a delimitar quines de les possibles solucions al problema són adequades (les que compleixen els requisits) i quines no*" (Pradel Miquel, J.; Raya Martos, J., 2011).

4.1 Especificació dels requisits

Els requisits d'aquesta aplicació són els que s'especifiquen en els objectius del TFG. El grau d'assoliment d'aquests objectius es correspondrà al grau de compliment amb els requisits de l'aplicació. Tanmateix en l'especificació dels requisits se sol distingir entre requisits funcionals i no funcionals, tot seguit es detallen quins són aquests per al TFG.

4.1.1 Requisits funcionals

"*Aquells que fan referència a la funcionalitat que ha de proporcionar el sistema [...] indiquen quin és el comportament del sistema davant dels estímuls que li arriben de l'exterior"* (Pradel Miquel, J.; Raya Martos, J., 2011).

1. Oferir un mecanisme automatitzat de recol·lecció periòdica de notícies anotades semànticament (etiquetades) de diferents fonts on-line.

2. Oferir un mecanisme normalitzador que permeti unificar els diferents vocabularis per tal d'emprar una semàntica única en l'etiquetatge de les notícies.

3. Definir algun mecanisme que permeti determinar a quina de les diferents accepcions fa referència el terme que descriu una etiqueta.

4. Categoritzar i classificar les etiquetes tot identificant el tipus d'ens a què es refereixen cadascuna d'elles.

5. Obtenir una representació lògica de les dades que permeti emmagatzemar les notícies, les etiquetes i els resultats de l'anàlisi en una mateixa base de dades independentment de la font de la que procedeixin les notícies.

6. Dur a terme algun procediment de descoberta de coneixements sobre les dades.

4.1.2 Requisits no funcionals

"*Aquells que fan referència a qualsevol altre requisit no relacionat amb la funcionalitat en sí del sistema"* (Pradel Miquel, J.; Raya Martos, J., 2011).

1. Es sistema base haurà d'operar com a mínim amb dues fonts de notícies.

2. Cal que la recol·lecció de notícies sigui un procés escalable i que no tingui límits de volum.

3. Oferir un entorn gràfic que permeti analitzar els resultats dels procediments anteriors així com visualitzar ràpidament quins són els temes majoritaris tractats en cada font d'informació.

4. Oferir el producte com una aplicació local que integri totes les funcionalitats anteriors.

5. Oferir el producte com a el conjunt de fitxers que integren el codi font i una sèrie de scripts que permetin compilar el programa així com generar un directori que pugui ser el distribuïble, juntament amb els scripts responsables de l'execució de l'aplicació.

6. Oferir els manuals adients per tal de dotar a l'usuari final de les instruccions per a compilar, executar i utilitzar el programa.

7. Documentar tot el projecte adientment.

4.2 Identificació dels casos d'ús

Els casos d'ús són una tècnica de documentació de requisits que permet fer servir diversos graus de detall i de formalisme i que està àmpliament estesa en el sector de l'enginyeria informàtica i suportada per l'estàndard UML. Tot seguit es detallen els casos d'ús detectats per a aquesta aplicació mitjançant el diagrama de casos d'ús i la seva descripció textual.

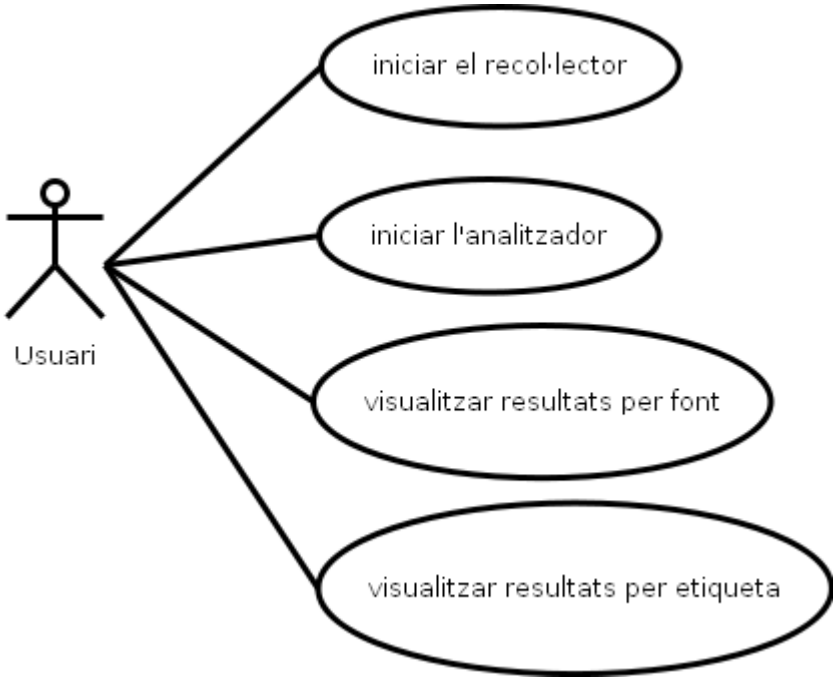


Figura 8 – Diagrama de casos d'ús de NewsAnalyzer

4.2.1 Descripció textual dels casos d'ús

Cas d'ús: **Iniciar el recol·lector**

- Actors: L'usuari de l'aplicació
- Objectius:
  - Iniciar el procediment automatitzat de recol·lecció de notícies etiquetades d'acord amb els requisits especificats per a l'aplicació.
  - Desar les notícies recol·lectades en una base de dades local per al seu posterior tractament per l'analitzador.
- Precondicions i garanties mínimes:
  - L'usuari haurà d'especificar a partir de quina data i de quines fonts vol iniciar la recol·lecta de notícies.
  - No es permetran valors no vàlids per als paràmetres de la recol·lecció.
  - La recol·lecta s'haurà de poder automatitzar (cada x minuts, diària o setmanal) i que no tingui límits de volum.
- Àmbit: La pròpia aplicació
- Escenari principal d'èxit:
  1) L'usuari indica que vol configurar/iniciar el recol·lector de notícies.
  2) L'usuari indica des de quina data vol que s'iniciï la recol·lecció.
  3) L'usuari indica de quines fonts cal recol·lectar notícies.
  4) L'usuari indica amb quina periodicitat vol que es recol·lectin les notícies a partir d'ara.
  5) L'usuari indica que vol que s'iniciï la recol·lecta
  6) Quan l'usuari està satisfet amb la quantitat d'articles recol·lectats, sol·licita que s'aturi la recol·lecció.

Cas d'ús: **Iniciar l'analitzador**

- Actors: L'usuari de l'aplicació
- Objectius:
  - Iniciar el procediment automatitzat de normalització i classificació de les etiquetes de les notícies que encara no hagin estat analitzades d'acord amb els requisits especificats per a l'aplicació.
  - Desar els resultats de l'anàlisi en la base de dades local.
- Precondicions i garanties mínimes:
  - El recol·lector haurà d'haver recol·lectat i desat les notícies a la base de dades i aquestes hauran d'estar marcades com a no analitzades.
  - L'usuari haurà d'especificar amb quina freqüència voldrà que es duguin a terme els anàlisis (aquests poden portar força estona).
- Àmbit: La pròpia aplicació
- Escenari principal d'èxit:
  1) L'usuari indica que vol configurar/iniciar l'analitzador.
  2) L'usuari indica amb quina freqüència vol que s'iniciï el procediment automatitzat d'anàlisi (normalització + classificació de les etiquetes).
  3) L'usuari indica que vol que s'iniciï l'anàlisi.
  4) Quan l'usuari està satisfet amb la quantitat d'articles analitzats, sol·licita que s'aturi l'anàlisi.

Cas d'ús: **Visualitzar els resultats per font**

- Actors: L'usuari de l'aplicació
- Objectius:
    - Mostrar la freqüència amb que es troben assignades cadascuna de les etiquetes normalitzades als articles de cadascuna de les fonts analitzades.
- Precondicions i garanties mínimes:
    - Hauran d'existir notícies a la base de dades i s'hauran d'haver analitzat.
- Àmbit: La pròpia aplicació
- Escenaris principal
    1) L'usuari indica que vol visualitzar els resultats.
    2) Se li mostra a l'usuari la vista sol·licitada.

Cas d'ús: **Visualitzar els resultats per etiqueta**

- Actors: L'usuari de l'aplicació
- Objectius:
    - Mostrar els resultats de la normalització i la classificació de les etiquetes a l'usuari.
- Precondicions i garanties mínimes:
    - Hauran d'existir notícies a la base de dades i s'hauran d'haver analitzat.
- Àmbit: La pròpia aplicació
- Escenaris principal
    1) L'usuari indica que vol visualitzar els resultats.
    2) Se li mostra a l'usuari la vista sol·licitada.

5 Fase d'anàlisi

"*L'anàlisi d'un sistema informàtic documenta com ha de ser el producte per desenvolupar des d'un punt de vista extern (és a dir, sense considerar com està fet per dintre). Habitualment, aquesta documentació es fa en forma de models*" (Pradel Miquel, J.; Raya Martos, J., 2011).

5.1 Model de domini

En el context de l'enginyeria del programari orientada a objectes, el model de domini és la representació en forma de classes de les entitats que representen el problema en el món real.

5.1.1 Diagrama UML estàtic de classes

El diagrama estàtic de classes resultant de l'anàlisi és el següent:

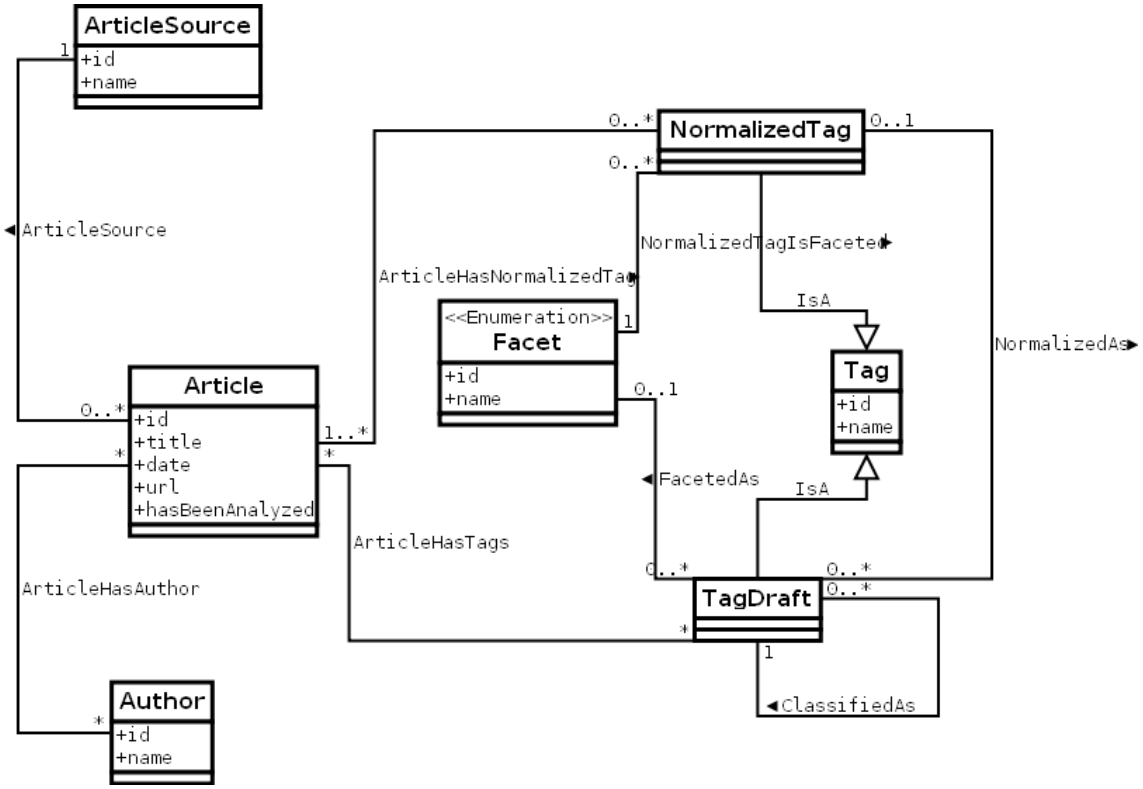


Figura 9 – Diagrama de classes del domini de NewsAnalyzer

Justificació

S'ha utilitzat un model simplificat ja que en una primera fase s'ha prioritzat que els algorismes principals de normalització i classificació funcionin. En un futur, si es volen ampliar les funcionalitats de l'aplicació, també es podrà ampliar fàcilment el model.

S'ha aïllat el que és l'article de les etiquetes (Tags) i en les etiquetes s'ha fet distinció entre dos tipus:

- TagDraft o esborrany d'etiqueta:

  Aquelles que assigna la font de les dades. Les he anomenat d'aquesta manera ja que cada font etiqueta les dades a la seva manera, sense emprar una nomenclatura estàndard.

  Aquest tipus d'etiqueta pot tenir un facet associat o no, en funció de si la font ofereix aquesta funcionalitat o no (The New York Times ho ofereix, The Guardian no).

  Un altres aspecte important és que sovint les fonts assignen una categoria a les etiquetes. Aquesta categoria es pot considerar com una etiqueta més i és interessant desar la relació de categorització entre etiquetes perquè pot servir en un futur per discernir entre els diferents significats que coincideixen amb el mot que identifica l'etiqueta.

- NormalizedTag o etiqueta normalitzada:

  Aquelles que assigna el programa durant el procediment de normalització. Aquestes etiquetes segueixen totes una mateixa nomenclatura.

  Aquestes etiquetes sempre tindran un facet assignat, ja que hauran estat classificades pel programa.

Els facets bàsics s'han categoritzat de la següent manera:

- Persona
  Exemples: bomber, Barack Obama...

- Organització
  Exemples: ajuntament, FBI...

- Lloc
  Exemples: Europa, Lleida...

- Instant o període en el temps
  Exemples: Edat Mitjana, any 2011

- Descriptor:
  Qualsevol etiqueta que no es pot categoritzar amb cap dels altres facets es pot considerar un simple descriptor. Diguem que la resta de facets deriven d'aquet.

Donat que els facets estan molt marcats, s'ha decidit implementar-los no com una classe sinó com una **Eumeració**.

5.2 Identificació de les classes frontera i de control i de les operacions.

Tot seguit es presentarà per a cada cas d'us un diagrama de col·laboració simplificat que mostra el resultat de la identificació de les classes frontera, de control i de les operacions que hi estan relacionades.

5.2.1 Diagrama de col·laboració del cas d'ús: iniciar el recol·lector

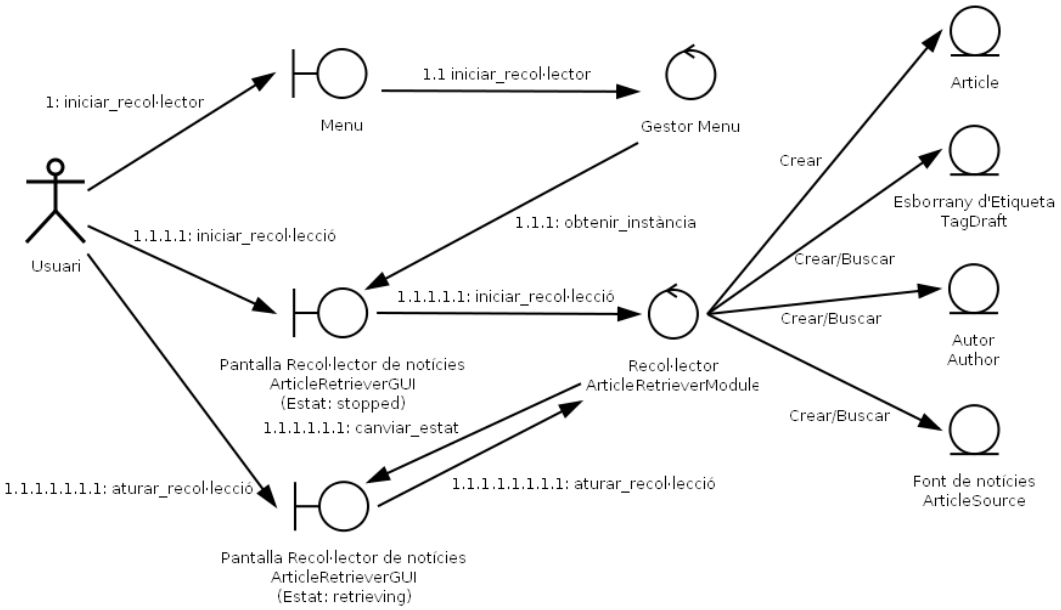


Figura 10 – Diagrama de col·laboració del cas d'ús "Iniciar el recol·lector"

Classes de frontera identificades: *Menú* i *ArticleRetrieverGUI*.
Classes de control identificades: *Gestor menú* i *ArticleRetrieverModule*.

5.2.2 Diagrama de col·laboració del cas d'ús: iniciar l'analitzador

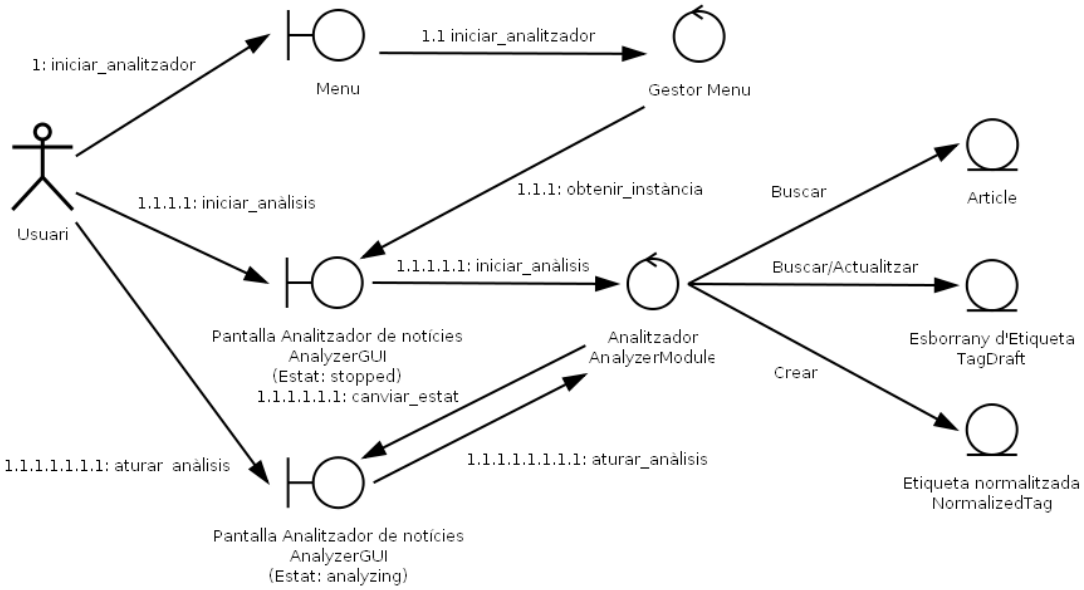


Figura 11 – Diagrama de col·laboració del cas d'ús "Iniciar l'analitzador"

Classes de frontera identificades: *Menú* i *AnalyzerGUI*.
Classes de control identificades: *Gestor menú* i *AnalyzerModule*.

5.2.3 Diagrama de col·laboració del cas d'ús: visualitzar resultats per font

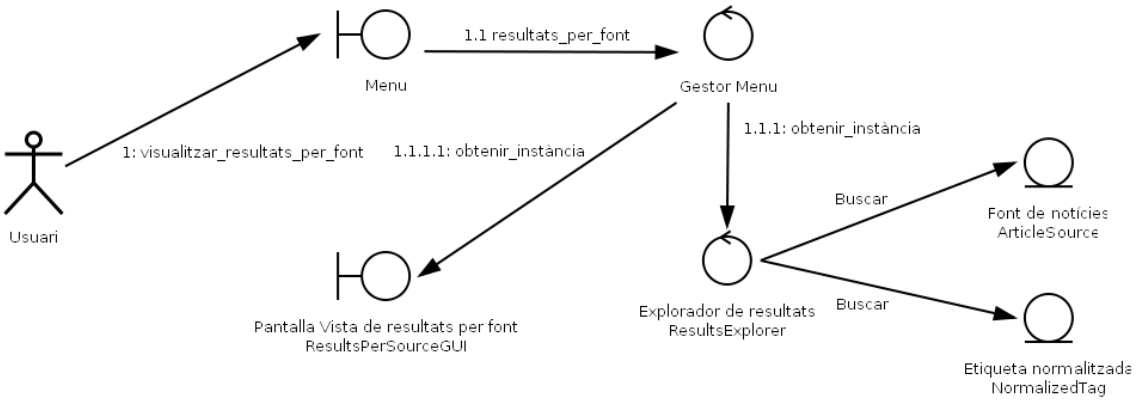


Figura 12 – Diagrama de col·laboració del cas d'ús "Visualitzar resultats per font"

Classes de frontera identificades: *Menú* i *ResultsPerSourceGUI*.
Classes de control identificades: *Gestor menú* i *ResultsExplorerModule*.

5.2.4 Diagrama de col·laboració del cas d'ús: visualitzar resultats per etiqueta

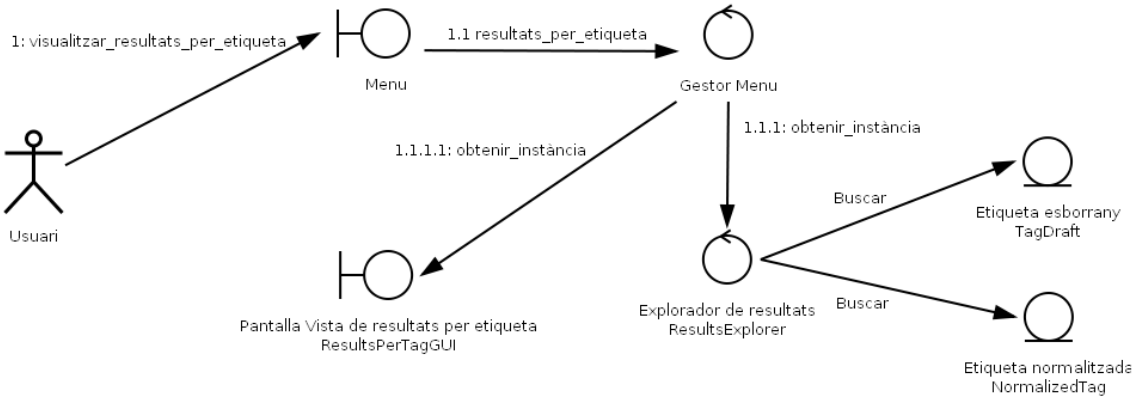


Figura 13 – Diagrama de col·laboració del cas d'ús "Visualitzar resultats per etiqueta"

Classes de frontera identificades: *Menú* i *ResultsPerTagGUI*.
Classes de control identificades: *Gestor menú* i *ResultsExplorer*.

5.3 Modelització de la Interfície Gràfica d'Usuari

Tot seguit es detalla la modelització de cadascuna de les classes frontera identificades en l'apartat anterior.

5.3.1 Pantalla Menu

Pantalla que permet seleccionar entre les diferents funcionalitats de l'aplicació.

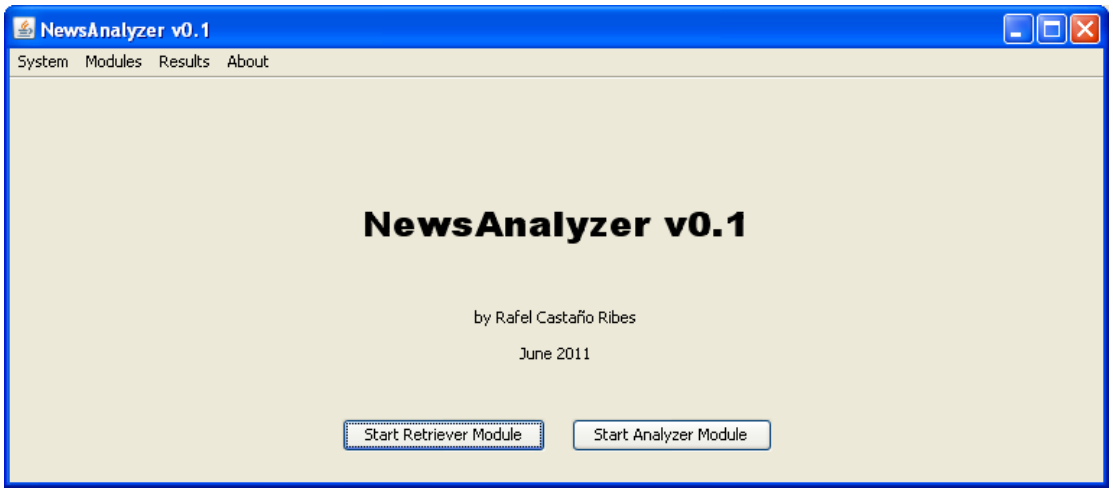


Figura 14 – Model de pantalla principal amb el menú de l'aplicació

Opcions del menú:

a) Sortir de l'aplicació

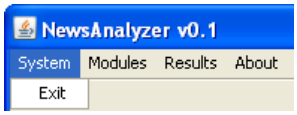


Figura 15 – Opcions del menú System

b) Iniciar els mòduls recol·lector i analitzador

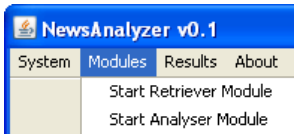


Figura 16 – Opcions del menú Modules

c) Mostrar els resultats de l'anàlisi per font i per etiqueta

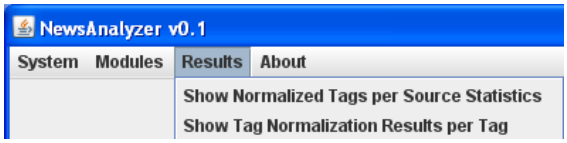


Figura 17 – Opcions del menú Results

d) mostrar informació general del programa

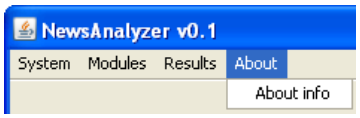


Figura 18 – Opcions del menú About

5.3.2 ArticleRetrieverGUI:

Pantalla que mostra i permet editar els paràmetres de configuració del mòdul recol·lector de notícies i permet iniciar o aturar la recol·lecció automàtica d'aquestes.

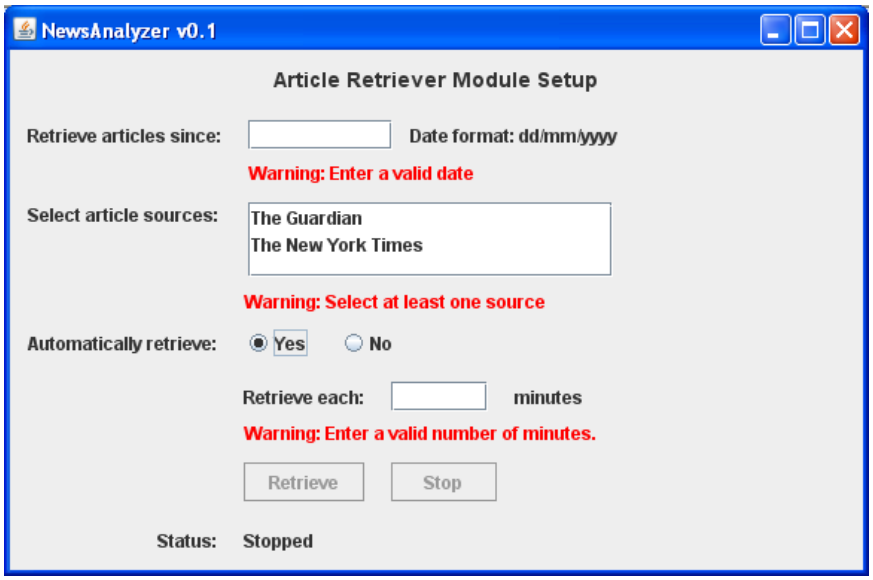


Figura 19 – Pantalla ArticleRetrieverGUI

5.3.3 AnalyzerGUI:

Pantalla que mostra i permet editar els paràmetres de configuració del mòdul analitzador i permet iniciar o aturar l'anàlisi automàtica de les notícies i etiquetes.

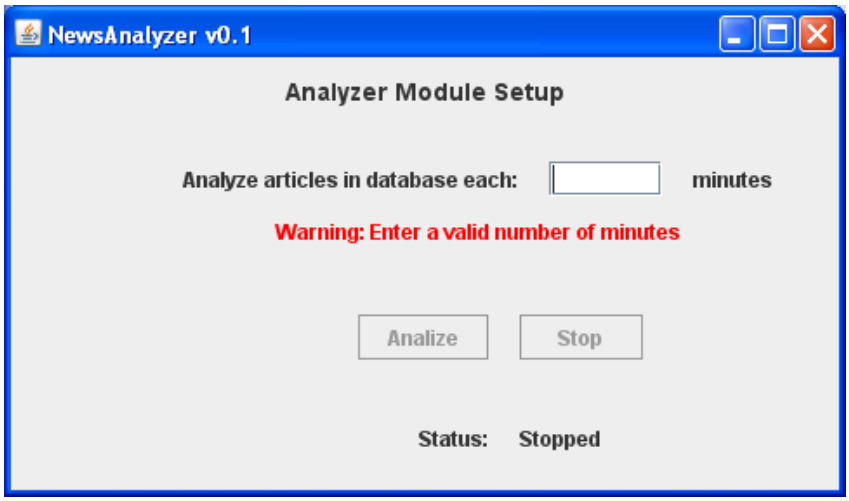


Figura 20 – Pantalla AnalyzerGUI

5.3.4 ResultsPerSourceGUI:

Pantalla que mostra la relació d'etiquetes normalitzades associades a cada font de notícies juntament amb la seva freqüència relativa.

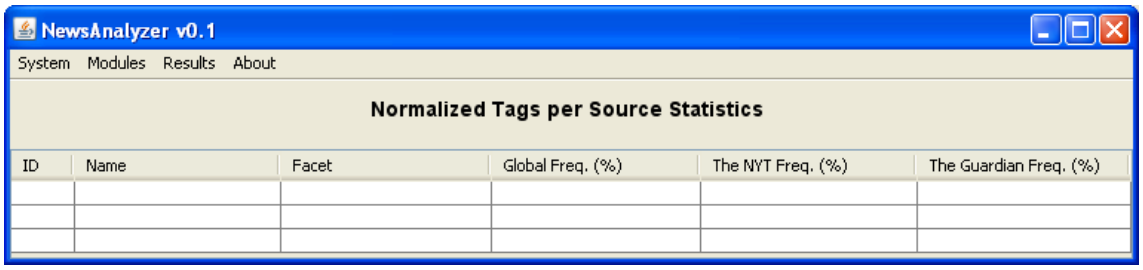


Figura 21 – Pantalla ResultsPerSourceGUI

5.3.5 ResultsPerTagGUI:

Pantalla que mostra el resultat de l'anàlisi per a cadascun dels tags esborrany que ha recol·lectat l'aplicació.

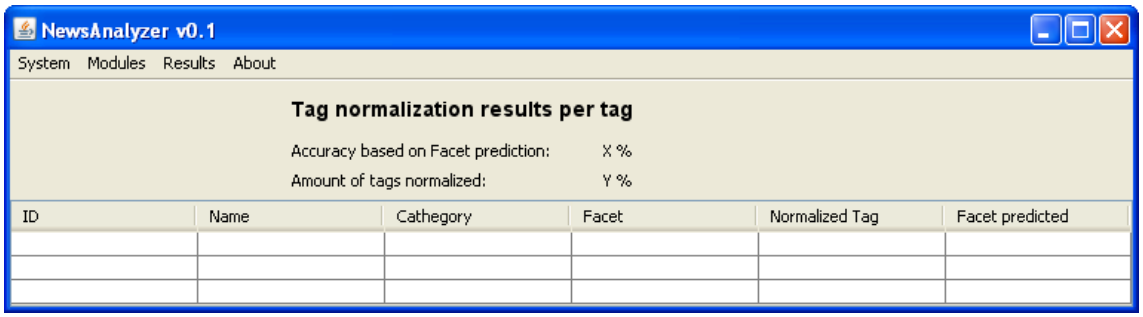


Figura 22 – Pantalla ResultsPerTagGUI

5.4 Relació de classes de control identificades

Tot seguit es detallen les classes de control identificades juntament amb la seva funcionalitat.

5.4.1 Gestor Menú:

S'encarregarà d'instanciar la pantalla corresponent a cada funcionalitat.

5.4.2 ArticleRetreiverModule

Aquesta classe serà instanciada per la pantalla ArticleRetrieverGUI i s'encarregarà de recollir els paràmetres de recol·lecció que hagi introduït l'usuari i d'invocar els mètodes corresponents de les classes corresponents per tal que s'iniciï la recol·lecció dels articles. Permetrà també aturar la recol·lecció automàtica si així ho sol·licita l'usuari des de la mateixa pantalla ArticleRetrieverGUI.

5.4.3 AnalyzerModule

Aquesta classe serà instanciada per la pantalla AnalyzerGUI i s'encarregarà de recollir els paràmetres d'automatització de l'anàlisi que hagi introduït l'usuari i d'invocar els mètodes corresponents de les classes corresponents per tal que s'iniciï l'anàlisi automàtic dels articles i etiquetes. Permetrà també aturar aquesta anàlisi automàtica si així ho sol·licita l'usuari des de la mateixa pantalla AnalyzerGUI.

5.4.4 ResultsExplorer

Aquesta classe serà instanciada pel propi menú i permetrà generar les taules que hauran de mostrar les pantalles ResultsPerSourceGUI i ResultsPerTagGUI.

6 Fases de disseny i d'implementació

*"La fase de **disseny** consisteix en documentar l'estructura i el comportament intern d'un sistema informàtic".*

*"La fase d'**implementació** consisteix en la creació del codi font".*

(Pradel Miquel, J.; Raya Martos, J., 2011)

Tot seguit es detallen els aspectes relacionats amb el disseny de NewsAnalyzer juntament amb els fragments de codi més rellevants que s'han utilitzat en la fase d'implementació.

6.1 Elecció del llenguatge de programació: Java

Donat que el llenguatge de programació principal utilitzat a les diferents assignatures del grau ha estat Java, per al desenvolupament d'aquest projecte s'ha decidit continuar amb aquesta tònica ja que els coneixements i l'experiència prèvia amb aquest llenguatge per part del desenvolupador són superiors als que pugui tenir amb qualsevol altre llenguatge de programació.

Concretament s'ha emprat el Java SE Development Kit 6 (JDK 6).

6.2 Elecció del sistema gestor de base de dades: MySQL

La base de dades s'ha decidit implementar en el sistema gestor de bases de dades MySQL donat que és un sistema de software lliure molt complert i amb el qual el programador ja hi havia treballat amb anterioritat, amb la qual cosa no calia dur a terme un nou aprenentatge, sinó únicament repassar els conceptes i comandes.

Concretament s'ha emprat el MySQL Community Server 5.5.8.

Per a dur a terme el disseny s'ha utilitzat el programa MySQL Workbench, també programari lliure i de la mateixa empresa, que ofereix una interfície gràfica que facilita molt la tasca de disseny i d'implementació.

Concretament s'ha emprat el MySQL Workbench 5.2.33

6.3 Elecció de la versió de WordNet

Donat que la versió 2.1 de WordNet és la més recent que es troba disponible tant per al sistema operatiu Microsoft Windows com per a sistemes Linux, s'ha decidit emprar aquesta versió.

Pàgina web oficial de WordNet: http://wordnet.princeton.edu/

6.4 Llibreries externes

Tot seguit es detallen quines han estat les llibreries externes emprades per tal de facilitar la feina de programació i d'interacció amb components externs al nucli del programa.

6.4.1 JDBC de MySQL

El JDBC (**J**ava **D**ata**b**ase **C**onnectivity) de MySQL és una llibreria que ofereix una API d'interacció amb el sistema gestor de bases de dades MySQL tot permetent l'execució de sentències SQL i altres comandes d'interès a través de codi Java.

Concretament s'ha emprat la versió 5.1.15 del JDBC de MySQL.

6.4.2 JSON simple

Les APIs dels diaris The New York Times i The Guardian retornen la informació sol·licitada dels articles en format JSON (JavaScript Object Notation). Existeixen un gran nombre de llibreries per permetre l'anàlisi sintàctica (parsing) d'aquest format des de Java. A la pàgina oficial[2] de JSON hi ha els enllaços a diverses d'aquestes llibreries. En aquest projecte s'utilitza la llibreria JSON.simple, per la seva senzillesa i facilitat d'ús i perquè té una llicència de programari lliure.

http://code.google.com/p/json-simple/

Concretament s'ha emprat la versió 1.1 d'aquesta llibreria.

6.4.3 Java API for WordNet Searching (JAWS)

La Java API for WordNet Searching (JAWS) és una llibreria que permet realitzar consultes mitjançant codi Java a l'aplicació WordNet. Cal però que WordNet estigui instal·lat en la màquina que executa l'aplicació i que s'hagin definit les variables de sistema que requereix la llibreria per tal de reconèixer en quin path es troba instal·lat el diccionari de WordNet.

La definició d'aquesta variable de sistema la realitza automàticament l'aplicació NewsAnalyzer mitjançant l'execució de la següent comanda Java:

```
String wnDbDir =
      NewsAnalyzerProperties.getInstance().getProperty(
      NewsAnalyzerProperties.WORDNET_DB_DIR );
System.setProperty("wordnet.database.dir", wnDbDir);
```

Ara bé, cal que l'usuari defineixi quina és la ruta cap al directori *dict* de WordNet en el fitxer NewsAnalyzerProperties.propoerties del directori base newsAnalyzer de l'aplicació:

[2] Pàgina web oficial de la notació JSON: http://www.json.org/

```
#The wordnet dict path
wordnet.database.dir=C:\\Program Files\\WordNet\\2.1\\dict
```

La pàgina web oficial d'aquesta API és http://lyle.smu.edu/~tspell/jaws/index.html.

Concretament s'ha emprat la versió 1.3 d'aquesta llibreria.

6.5 Disseny de la persistència

En el disseny de la persistència per a una base de dades relacional cal tenir en compte dos aspectes fonamentals: proporcionar la transformació del model de negoci obtingut a la fase d'anàlisi a un model relacional, i determinar el mecanisme mitjançant el qual els objectes seran emmagatzemats a la base de dades.

6.5.1 Model Relacional

El model relacional de les dades es defineix mitjançant el diagrama entitat-relació, que representa la base de dades que permetrà desar les estructures de dades del diagrama estàtic d'anàlisi. Per a NewsAnalyzer el diagrama obtingut ha estat el següent:

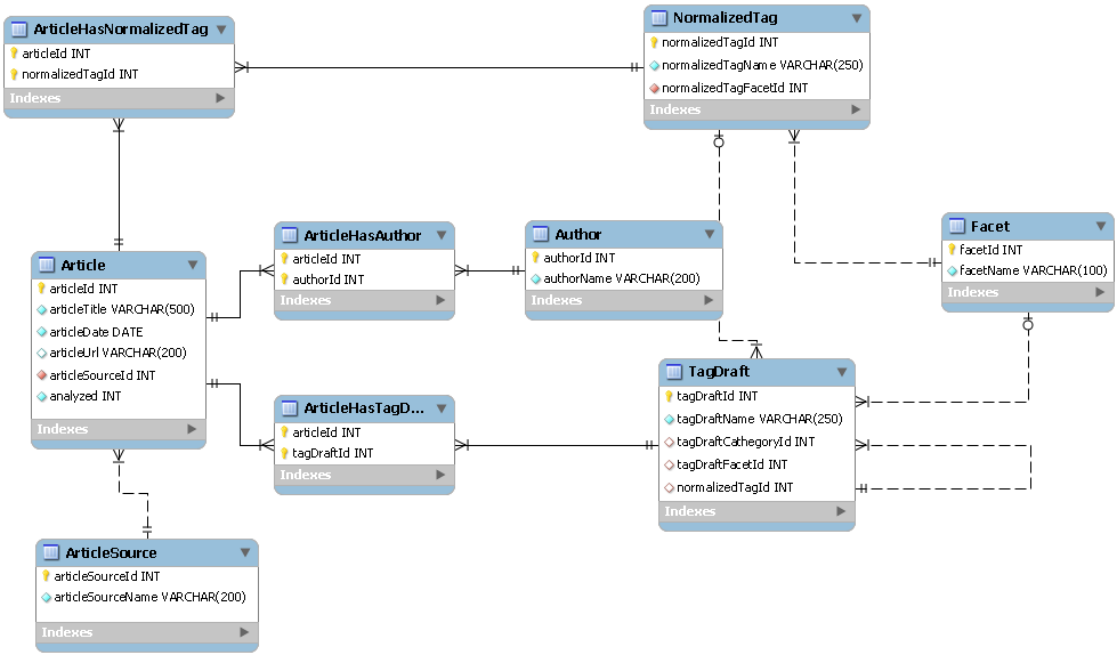


Figura 23 – Diagrama Entitat-Relació de NewsAnalyzer

Tot seguit s'explicaran amb una mica més de detall cadascuna de les taules:

ArticleSource: font d'articles

- articleSourceId: serà la **clau primària**, és un identificador enter sense signe amb valor **autoincremental**.
- articleSourceName: és el nom de la font d'articles. És **clau alternativa** i no pot ser nul.

Author: autor

- authorId: serà la **clau primària**, és un identificador enter sense signe amb valor **autoincremental**. No pot ser nul.
- authorName: és el nom de l'autor. És **clau alternativa** i no pot ser nul.

Article: article

- articleId: serà la **clau primària**, és un identificador enter sense signe amb valor autoincremental. No pot ser nul.
- articleTitle: el títol de l'article. No pot ser nul.
- articleDate: la data de publicació de l'article. No pot ser nul.
- articleUrl: si l'article té una URL, aquest camp la conté. Pot ser nul.
- articleSourceId: **clau forana** que referencia la font a què pertany aquest article. No pot ser nul.
- analyzed: enter que prendrà el valor 0 si l'article no ha estat analitzat encara i 1 si ja ho ha estat. El valor per defecte d'aquest atribut és 0. No pot ser nul.

ArticleHasAuthor: relació N:M d'autoria entre articles i autors

- articleId: **clau forana** que referencia a un article
- authorId: **clau forana** que referencia un autor.
- Tots dos camps formen la **clau primària** i no poden ser nuls.

Facet: la relació de facets

- facetId: enter sense signe que identifica al facet. És **clau primària**. El seu valor no és automàtic sinó que es determina a l'insertar els facets, fet que es produeix a l'instal·lar la base de dades. No pot ser nul.
- facetName: nom del facet. És **clau alternativa**. No pot ser nul.

NormalizedTag: etiqueta normalitzada i classificada

- normalizedTagId: serà la **clau primària**, és un identificador enter sense signe amb valor **autoincremental**. No pot ser nul.
- normalizedTagName: el nom de l'etiqueta normalitzada. No pot ser nul.
- normalizedTagFacetId: **clau forana** al facet amb el qual es relaciona l'etiqueta normalitzada. No pot ser nul.
- normalizedTagName i normalizedTagFacetId formen una **clau alternativa**, és a dir, no hi pot haver dues etiquetes normalitzades amb el mateix nom i el mateix facet assignat.

ArticleHasNormalizedTag: relació N:M entre articles i etiquetes normalitzades. Aquesta taula no seria necessària, ja que aquesta relació es podria obtenir a partir del camp normalizedTagId de la combinació de TagDraft amb ArticleHasTagDrat, però es manté aquesta taula per qüestions d'eficiència en les consultes a la base de dades.

- articleId: **clau forana** que referencia a un article
- normalizedTagId: **clau forana** que referencia a una etiqueta normalitzada.
- Tots dos camps formen la **clau primària** i no poden ser nuls.

TagDraft: esborrany d'etiqueta (etiqueta original)

- tagDraftId: serà la **clau primària**, és un identificador enter sense signe amb valor **autoincremental**. No pot ser nul.
- tagDraftName: el nom de l'etiqueta original

- tagDraftCathegoryId: **clau forana** que referencia al TagDraft del qual dervia aquest. Pot ser nul.
- tagDraftFacetId: **clau forana** que referencia el facet que té assignat aquesta etiqueta per part de la font. Pot ser nul.
- normalizedTagId: **clau forana** que referencia el tag normalitzat que li ha estat assignat a aquesta etiqueta per part d'aquest programa.

ArticleHasTagDraft: relació N:M entre articles i esborranys d'etiquetes.

- articleId: **clau forana** que referencia a un article
- tagDraftId: **clau forana** que referencia a un esborrany d'etiqueta.
- Tots dos camps formen la **clau primària** i no poden ser nuls.

6.5.2 Gestió de la persistència (package *dataModel.DAOs*)

Per dur a terme l'accés a les dades s'ha pres com a referència el patró de disseny DAO (Data Acces Object[3]) perquè és el que es va explicar en l'assignatura d'Enginyeria del Programari Orientat a Objectes. Si bé hi ha bastions actuals que permeten un accés a les dades a més alt nivell, el temps d'aprenentatge d'aquests és força elevat i donat que el temps del TFG és molt ajustat no s'han tingut en consideració.

L'estructura bàsica de la gestió de la persistència és la següent:

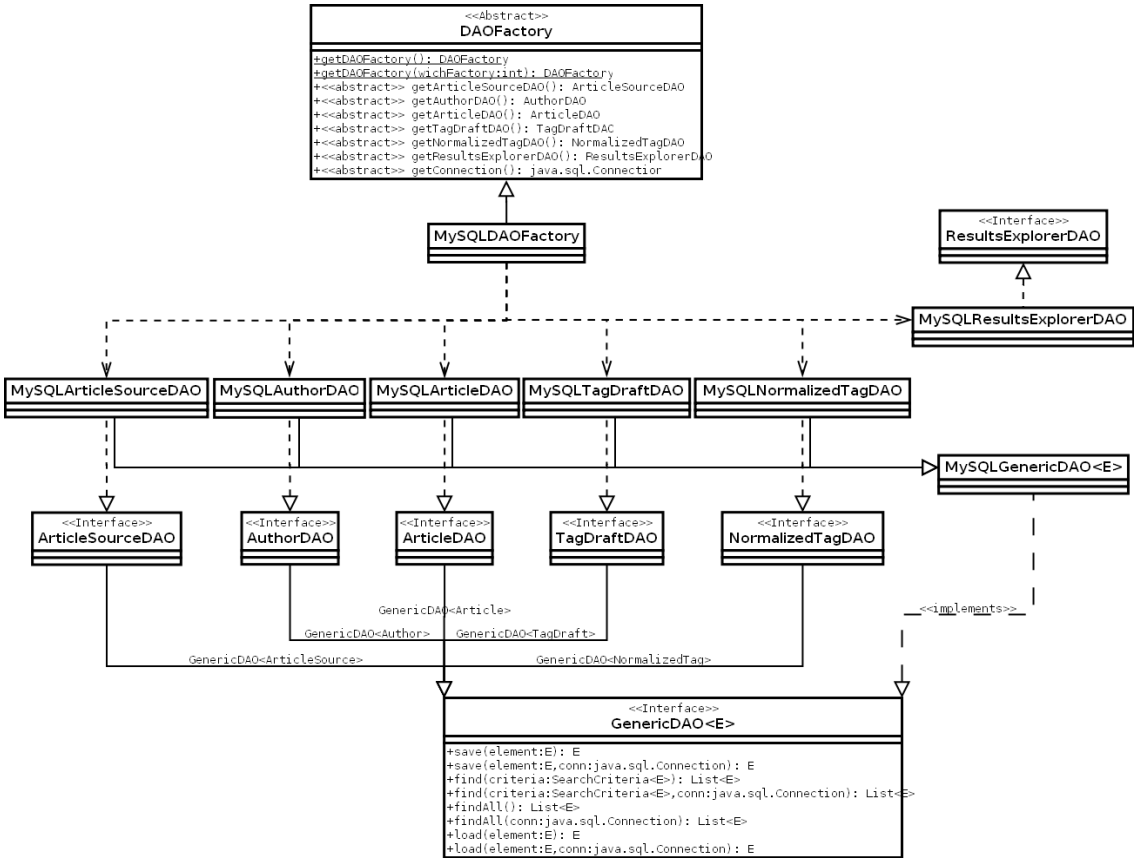


Figura 24 – Implementació del patró DAO: diagrama de classes resultant

[3] Més informació a http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html

Per tal d'aïllar la gestió de la persistència del sistema gestor de bases de dades (SGBD) concret a utilitzar, el patró DAO utilitza dos mètodes fonamentals:

- En primer lloc cal definir una interfície DAO amb els mètodes necessaris per a cada entitat per la qual s'ha de gestionar la persistència. Per a cada SGBD concret i cada interfície caldrà crear una classe que implementi aquests mètodes per a aquest SGBD concret.

- En segon lloc cal crear una fàbrica de DAOs abstracta (Factory) que obliga a les seves classes derivades a implementar l'obtenció de cadascun dels DAOs per al SGBD concret, així com el mètode per a obtenir una connexió a la base de dades. A més a més, aquesta fàbrica abstracta ha de tenir un mètode estàtic de classe que permeti que es puguin instanciar objectes de les implementacions derivades d'aquesta.

Per tal de facilitar la tasca d'accés a la base de dades, donat que hi ha molta redundància entre mètodes dels diferents DAOs, s'ha utilitzat la potència que tenen les classes parametritzades de Java i s'ha creat la interfície GenericDAO<E> i la classe MySQLGenericDAO<E> que implementa aquesta interfície.

Això ha permès generalitzar els procediments de desar, seleccionar i carregar objectes i facilitar el desenvolupament d'aquest paquet de persistència.

Convé destacar que s'entén en aquest treball per desar, seleccionar i carregar:

- S'entén per **desar (save)** el fet de desar l'objecte a la base de dades si aquest no existeix o bé actualitzar-lo si ja hi existeix.

- S'entén per **seleccionar (find)** obtenir el llistat d'objectes que compleixen amb uns criteris de cerca determinats. Per tal de definir aquests criteris de selecció s'han creat un conjunt de classes especials que deriven totes d'una classe parametritzada anomenada SearchCriteria<E>.

- S'entén per **carregar (load)** el fet d'obtenir un representant d'un objecte de la següent manera: si ja existeix l'objecte a la base de dades, se selecciona aquest i es retorna a l'aplicació; si l'objecte no existeix a la base de dades, es desa i s'obtenen les dades actualitzades d'aquest per tal de retornar-lo a l'aplicació.

La classe parametritzada MySQLGenericDAO<E> té tot un seguit de mètodes d'obligada implementació en les seves classes derivades. Aquests mètodes són, entre altres els que permeten obtenir les sentències SQL per a executar les funcions, omplir-les amb els paràmetres corresponents, obtenir els criteris de cerca (SearchCriteria<E>), dur a terme les accions prèvies o posteriors a una inserció, actualització o consulta, etc.

La classe SearchCriteria<E> funciona d'una forma molt senzilla. Permet carregar un objecte com a criteri de cerca, però els atributs que admeten atributs nuls necessiten un tractament especial, ja que cal indicar si el que es desitja és cercar aquell objecte on aquell camp té valor null o bé si volem obtenir tots els objectes amb independència del valor que tingui aquell camp.

Per això, els objectes derivats de SearchCriteria<E> han d'inicialitzar el vector booleà searchWithIsNullField amb el número de camps que admeten valor nul i definir una sèrie de constants numèriques d'accés públic per tal que hom es pugui referir a aquestes posicions del vector.

A tall d'exemple es mostra el codi de la classe TagDraftCriteria, que permet obtenir criteris de cerca per als esborranys d'etiquetes:

```
public final static int CATHEGORY = 0;
public final static int FACET = 1;
public final static int NORMALIZED_TAG = 2;

private void initialize(){
    searchWithIsNullField = new boolean[3];
    for(int i=0; i<this.searchWithIsNullField.length; i++)
        this.searchWithIsNullField[i]=true;
}

public TagDraftCriteria(){
    super();
    initialize();
}
```

6.6 Disseny del mòdul recol·lector de notícies (package *retrievers*)

El mòdul recol·lector de notícies és aquell que s'encarrega de recol·lectar notícies a través de les APIs de les diferents fonts disponibles.

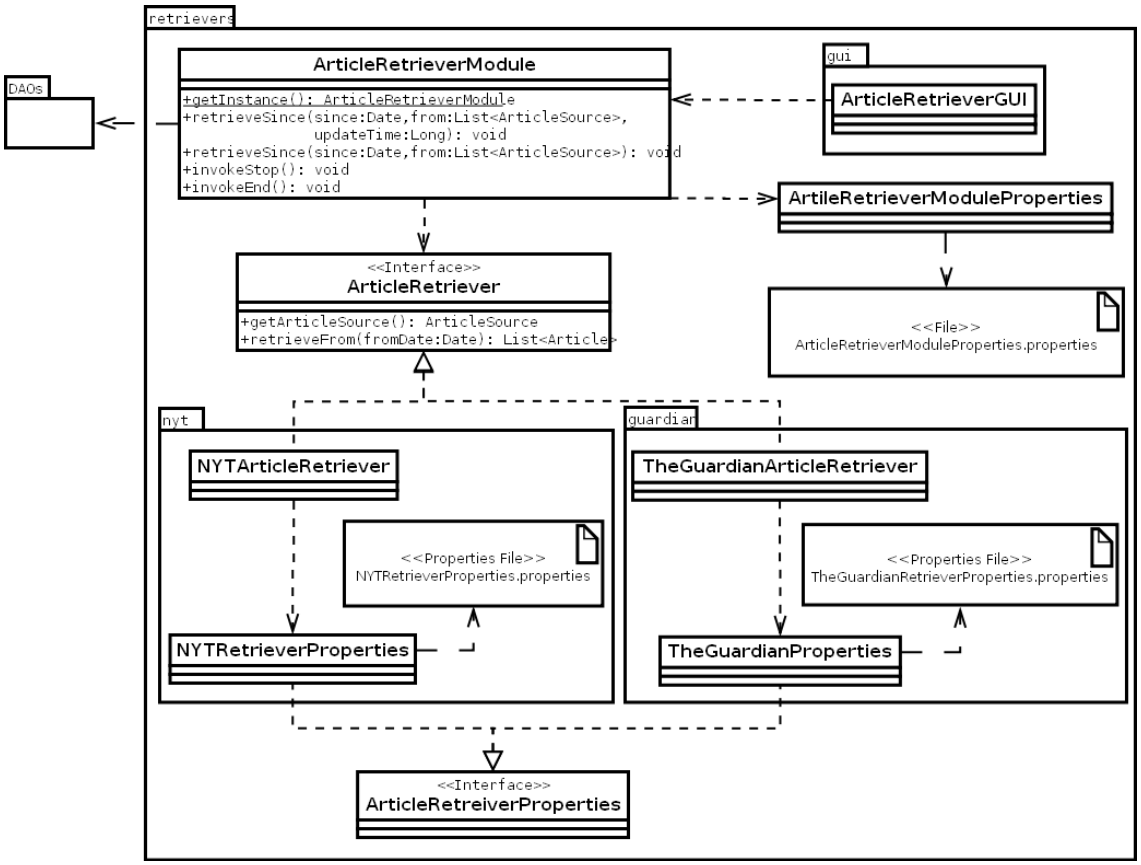


Figura 25 – Diagrama UML del mòdul recol·lector (retrievers)

Aquest mòdul s'estructura de forma modular de tal manera que per a cada font de notícies cal instal·lar un recol·lector específic. Això és així perquè cada font disposa de la seva pròpia API i aquestes no segueixen cap tipus d'estàndard, amb la qual cosa cada recol·lector acostuma a implementar un paradigma diferent.

La implementació d'aquest mòdul es troba en el package *retrievers*, que consta de les següents classes, subpackages i fitxers:

- Classe: ArticleRetrieverModule

  Classe **controladora** del mòdul que s'encarrega d'interaccionar amb cadascun dels recol·lectors específics i oferir els seus serveis a la Interfície Gràfica d'Usuari.

  Per tal d'impedir que la GUI hagi d'esperar a què acabi l'execució dels mètodes d'aquesta classe, i per tal que l'execució dels mètodes es pugui aturar en qualsevol moment, aquesta classe és una derivada de Thread, i disposa dels mètodes invokeStop i invokeEnd per tal d'aturar l'execució dels mètodes o del propi Thread respectivament. Encara és en desenvolupament.

  En instanciar-se aquesta classe carrega diferents mòduls de recol·lectors de notícies que s'hagin predefinit en el fitxer ArticleRetrieverModuleProperties.properties. Això es duu a terme amb l'ajuda de la classe auxiliar ArticleRetrieverModuleProperties.
- Fitxer: ArticleRetrieverModuleProperties.properties

  Fitxer de text en el qual s'indica el nom i subpackage de la classe responsable de carregar les propietats cada submòdul recol·lector.

  Cal emprar el següent format:

  ```
  NomDelSubmòdul=NomDelSubpackage.NomDeLaClasseProperties
  ```

- ArticleRetrieverModuleProperties.java

  Classe auxiliar que carrega el fitxer ArticleRetrieverModuleProperties.properties i el presenta a la classe ArticleRetrieverModule.

- ArticleRetriever.java

  Interfície que han d'implementar els diferents recol·lectors específics. Conté la definició de les funcions bàsiques de què ha de disposar cadascun dels recol·lectors per tal de poder ésser utilitzats per la classe ArticleRetrieverModule.

- ArticleRetrieverProperties.java

  Interfície que han d'implementar les classes responsables de la càrrega de les propietats de cada recol·lector per tal que es puguin instanciar aquestos des de la classe ArticleRetrieverModule.

- Subpackages *nyt* i *guardian*

Són dos mòduls recol·lectors instal·lats per defecte en l'aplicació que permeten recol·lectar notícies per als diaris The New York Times i The Guardian respectivament. Els detalls d'implementació d'aquestos es detallen més endavant.

- Subpackage *gui*

  Es tracta de la interfície gràfica d'usuari del mòdul, es detalla més endavant.

6.6.1 Recol·lectors

En el package *retrievers*, cada recol·lector específic es desenvolupa com un subpackage.

Per desenvolupar un recol·lector específic cal seguir els següents passos:

1) Crear la classe responsable de la recol·lecció, que ha d'implementar la interfície ArticleRetriever.

   El mètode bàsic és *List<Article> retrieveFrom(Date date)*, que retorna el llistat d'articles de la data definida en el paràmetre *date*. **Cal però que aquest mètode es responsabilitzi de consultar a la base de dades quins articles ja s'han descarregat per tal de no tornar-los a descarregar de nou**!

   Un altre mètode important és *ArticleSource getArticleSource()*, que retorna un objecte de tipus ArticleSource que representa la font de dades amb què treballa aquest recol·lector. Aquest mètode és necessari per tal que la GUI pugui mostrar els recol·lectors disponibles.

2) Crear el fitxer .properties corresponent que contindrà totes aquelles propietats necessàries per a interaccionar amb l'API de la font concreta. Per exemple, en el cas del The New York Times és necessari registrar-se en el seu aplicatiu web i obtenir una clau per tal de tenir accés als articles, doncs en el fitxer .properties hi ha la clau obtinguda per a aquest aplicatiu.

   Els fitxers .properties són fitxers de text en els quals els diferents paràmetres s'especifiquen amb el següent format:

   ```
   nomDelParàmetre=valor
   ```

   Dos paràmetres importants que han de constar en aquest fitxer són el nom de la font de notícies i el nom de la classe del punt 1). El nom d'aquests paràmetres ha de ser, respectivament, ***ArticleSourceName*** i ***MainClass***.

3) Crear la classe responsable de carregar el fitxer .properties definit en l'apartat 2). Aquesta classe ha d'implementar la interfície ArticleRetrieverProperties, on el mètode principal és *String getMainClass()*, que permet a la classe ArticleRetrieverModule carregar la classe principal definida en l'apartat 1).

Tot seguit es comenta l'estructura i funcionament de cadascun dels mòduls que es troben implementats en la versió actual de l'aplicació.

6.6.2 The New York Times (subpackage *nyt*)

Estructura del mòdul

- retrievers.nyt
  - NYTArticleRetriever: Classe principal
  - NYTRetrieverProperties.java: Classe de càrrega del fitxer .properties
  - NYTRetrieverProperties.properties: Fitxer .properties

Requisits

Es requereix una **clau d'accés** per accedir a l'API del The New York Times. Aquestes claus les proporciona el propi diari i requereixen d'un registre previ per part de l'usuari. NewsAnalyzer disposa d'una clau registrada per al desenvolupament i prova de l'aplicació.

Si l'usuari vol canviar la clau que utilitza NewsAnalyzer per una de pròpia ho pot fer al fitxer NYTRetrieverProperties.properties, canviant el valor de la propietat NYTAPIKey.

L'adreça web per obtenir noves claus és:

http://developer.nytimes.com/docs/reference/keys

Funcionament

Com ja s'ha comentat amb anterioritat, l'API del New York Times (NYT) permet la consulta d'articles a traves de sol·licituds HTTP get. El resultat és retornat en format JSON i de forma paginada, amb els articles ordenats per instant de publicació de més antics a més recents i deu articles per pàgina.

La consulta bàsica que du a terme aquest mòdul és la següent:

```
http://api.nytimes.com/svc/search/v1/article
      ?format=json
      &query=date:[DATE] classifiers_facet:[Top/News]
      &fields=title, url, byline, classifiers_facet,
              des_facet, geo_facet, per_facet, org_facet
      &offset=OFFSET
      &api-key=API_KEY
```

Els paràmetres emprats en aquesta consulta han estat els següents:

- **format**: per determinar el format de la resposta. JSON és l'únic disponible de moment
- **query**: la consulta:
  - **date**: la data de la qual es volen obtenir els articles en format yyyyMMdd.

    - **classifiers_facet**: filtre que permet delimitar quin tipus d'articles descarregar, en el cas d'aquesta consulta, únicament es descarreguen notícies (classificats, mots encreuats i altres tipus d'articles queden descartats).
- **fields**: camps que es volen obtenir en la resposta. Els camps escollits han estat el títol (title), l'adreça web de la notícia (url), els autors (byline), les etiquetes de classificació (classifiers_facet), les etiquetes corresponents a cada facet:
    - **des_facet**: identifica una etiqueta descriptiva
    - **geo_facet**: identifica una localització geogràfica
    - **per_facet**: identifica una persona o tipus de persona
    - **org_facet**: identifica una organització
- **offset**: pertmet indicar quina pàgina de resultats es vol obtenir (recordi's que es retornen en pàgines de 10 articles, de menys a més recents).
- **api-key:** permet proporcionar la clau d'accés a l'API.

La documentació per interpretar els paràmetres de la cerca es troba a la pàgina web:

http://developer.nytimes.com/docs/article_search_api/

Es pot provar la consulta a la consola de proves de l'API a:

http://prototype.nytimes.com/gst/apitool/index.html

El resultat de la consulta s'obté en format JSON. Aquest mòdul utilitza la llibreria JSON-Simple per parsejar el resultat i convertir el text en objectes de tipus Article i TagDraft.

Més informació sobre aquesta llibreria a:

http://code.google.com/p/json-simple/

En cas de produir-se algun error en la descàrrega dels articles, els codis d'error que retorna l'API del NYT es troben a:

http://developer.nytimes.com/docs/reference/errors

Limitacions

Aquesta API té una limitació de **5000** sol·licituds per clau i dia, amb la qual cosa cal anar amb compte de no esgotar-les.

6.6.3 The Guardian (subpackage guardian)

Estructura del mòdul

- retrievers.guardian
    - TheGuardianRetrieverProperties.java
    - TheGuardianRetrieverProperties.properties
    - TheGuardianArticleRetriever

Requisits

Tot i que es poden obtenir claus d'accés per a obtenir funcionalitats addicionals de l'API, per a aquesta aplicació no són necessàries.

Funcionament

A l'igual que el NYT, l'API del The Guardian permet la consulta d'articles a traves de sol·licituds HTTP get. El resultat és retornat en format JSON o XML, de forma paginada amb deu articles per pàgina, i amb els articles ordenats de la manera que s'indiqui.

La consulta bàsica que du a terme aquest mòdul és la següent:

```
http://content.guardianapis.com/search
      ?tag=type/article,tone/news
      &from-date=DATE
      &to-date=DATE
      &format=json
      &show-tags=keyword,contributor
      &page=PAGE
      &order-by=oldest
```

Els paràmetres emprats en aquesta consulta han estat els següents:

- **tag**: filtre que permet delimitar quin tipus d'articles descarregar, en el cas d'aquesta consulta, únicament es descarreguen articles i concretament de tipus notícies (classificats, mots encreuats i altres tipus d'articles queden descartats).
- **from-date**: la data des la qual es vol iniciar la recol·lecta de notícies, en format yyyy-MM-dd.
- **to-date**: la data fins a la qual es vol recol·lectar notícies, en format yyyy-MM-dd.
- **format**: per determinar el format de la resposta. JSON és l'únic disponible de moment
- **show-tags**: camps addicionals que es volen obtenir en la resposta. Els camps escollits han estat les paraules clau (keyword), que són les etiquetes descriptores assignades a la notícia, i els autors (contributor). A més a més, per defecte s'obtenen altres camps d'interès com el títol (webTitle), la url (webUrl) i el nom de la categoria (sectionName).
- **page**: permet indicar quina pàgina de resultats es vol obtenir (recordi's que es retornen en pàgines de 10 articles).
- **order-by:** permet indicar l'ordre amb què es volen obtenir els articles.

La documentació per interpretar els paràmetres de la cerca es troba a la pàgina web:

http://www.guardian.co.uk/open-platform/content-api-content-search-reference-guide

Es pot provar la consulta a la consola de proves de l'API a:

http://explorer.content.guardianapis.com/#search-endpoint-tab

El resultat de la consulta s'obté en format JSON. A l'igual que en el cas de l'API del NYT aquest resultat es parseja mitjançant la llibreria JSON-Simple i es converteix en objectes de tipus Article i TagDraft.

6.6.4 Interfície gràfica d'usuari (subpackage gui)

La interfície gràfica d'usuari (GUI) d'aquest mòdul consta de la classe ArticleRetreiverGUI, que deriva de JFrame, i utilitza la classe ArticleRetrieverModule per tal de desenvolupar la seva funcionalitat. El funcionament de la GUI es troba detallat en el manual d'usuari de l'aplicació.

6.6.5 Aspectes que han quedat pendents

Un problema que encara no s'ha resolt en els recol·lectors és com gestionar la diferència en el fus horari. Així, quan es demana recol·lectar notícies des del dia 04/06/2011 aquesta és la data que es pren com a referència per a cada fus horari. Per tant, com que als Estats Units l'hora està desfasada en -6h respecte a l'hora a Espanya, a la 01:00h de la matinada espanyola el recol·lector no trobaria cap notícia publicada a Estats Units i no es rebrien els articles fins a les 06:00h.

No és un problema difícil de resoldre, però degut a la manca de temps s'ha deixat per a un futur la seva resolució.

Un altre aspecte pendent és la GUI, que es troba en una versió molt incipient i s'ha de continuar desenvolupant. Per exemple, la lògica de l'aplicació està preparada per recol·lectar les notícies d'un dia determinat, però no la GUI.

6.7 Mòdul analitzador (package analyzer)

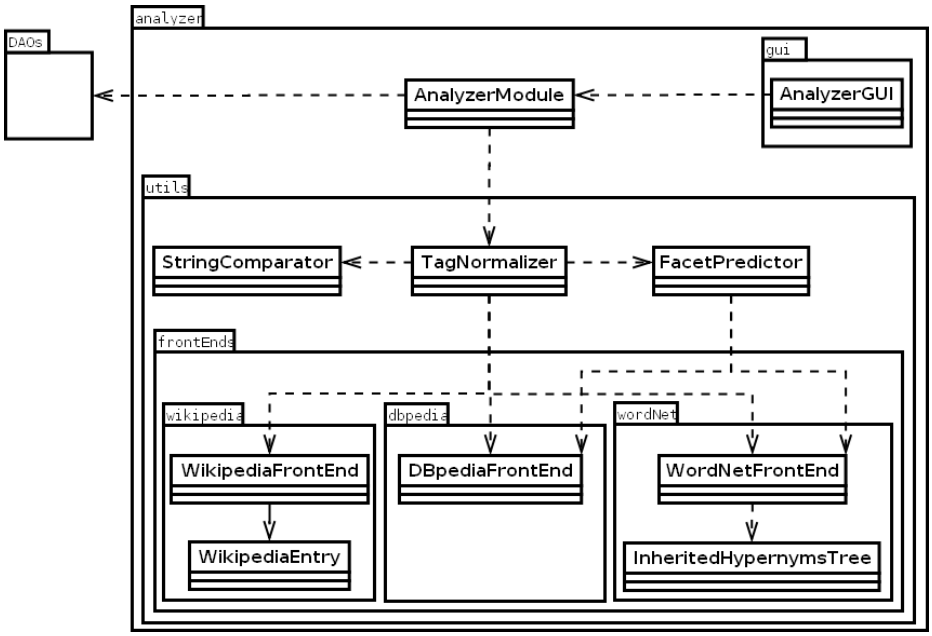


Figura 26 – Diagrama UML del mòdul analitzador (analyzer)

Aquest mòdul s'encarrega de fer un anàlisi de les notícies recol·lectades i assignar a cada notícia i etiqueta original una etiqueta normalitzada per tal de poder fer comparacions entre diferents notícies i fonts.

Les classes d'aquest mòdul es troben en el package *analyzer*. Aquest conté també dos subpackages que són el subpackage *utils*, que conté tot un seguit de classes sobre les quals es delegaran determinats aspectes dels anàlisis, i el subpackage *gui*, que contindrà les classes de la interfície gràfica d'usuari (GUI).

En la figura següent es detalla el diagrama UML de packages i classes d'aquest mòdul i acte seguit es detalla el contingut dels subpackages.

6.7.1 Classe AnalyzerModule

La classe controladora d'aquest mòdul és AnalyzerModule, que a l'igual que la controladora del mòdul recol·lector és una classe derivada de Thread i, per tant, permet l'execució i aturada no modals de mètodes per part del la GUI.

El principal mètode d'aquesta classe és *analyzeNotAnalyzedArticles*(long t):void. Aquest mètode executa periòdicament (cada t mil·lisegons) la funció que es detalla en el següent diagrama d'activitat:

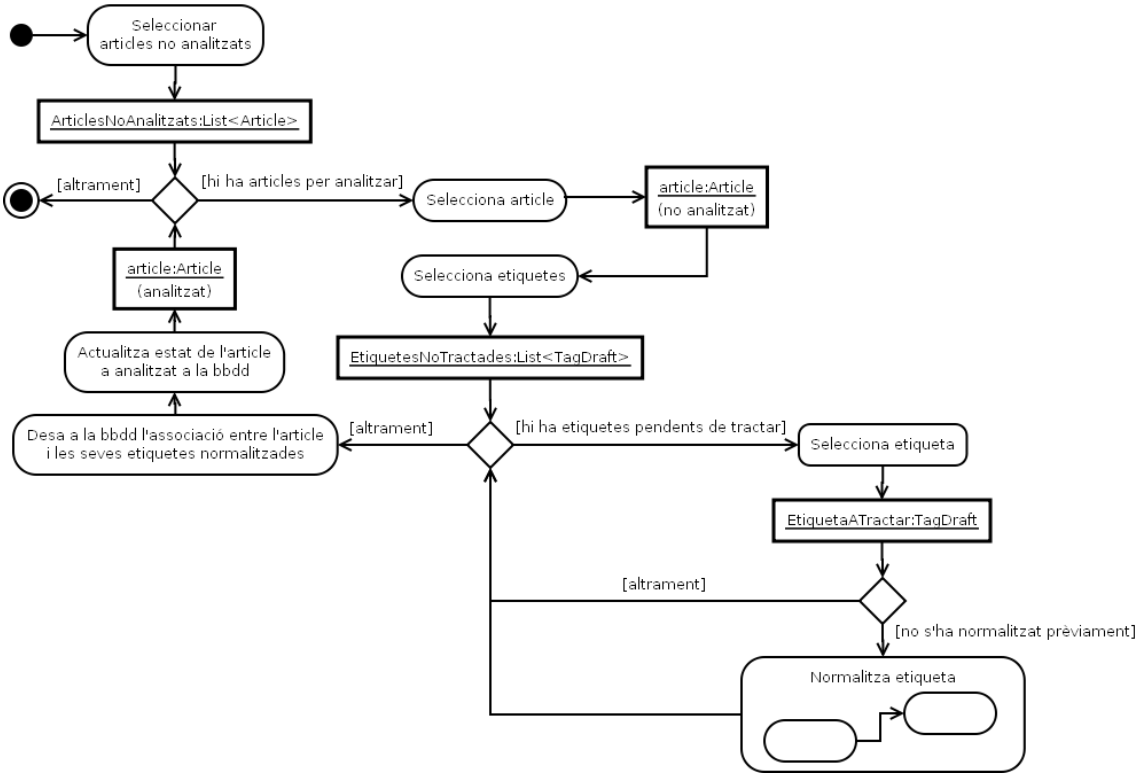


Figura 27 – Diagrama d'activitat del mètode principal d'AnalyzerModule

L'activitat *normalitza etiqueta* es delega a la classe *TagNormalizer*, del subpackage *utils*.

6.7.2 Submòdul d'utilitats (subpackage *utils*)

Aquest subpackage és probablement el més important de tota l'aplicació. En ell s'hi troben les classes d'utilitat que permeten dur a terme la normalització i classificació de les etiquetes.

El següent diagrama d'activitats descriu com es du a terme aquest procediment en l'aplicació newsAnalizer:

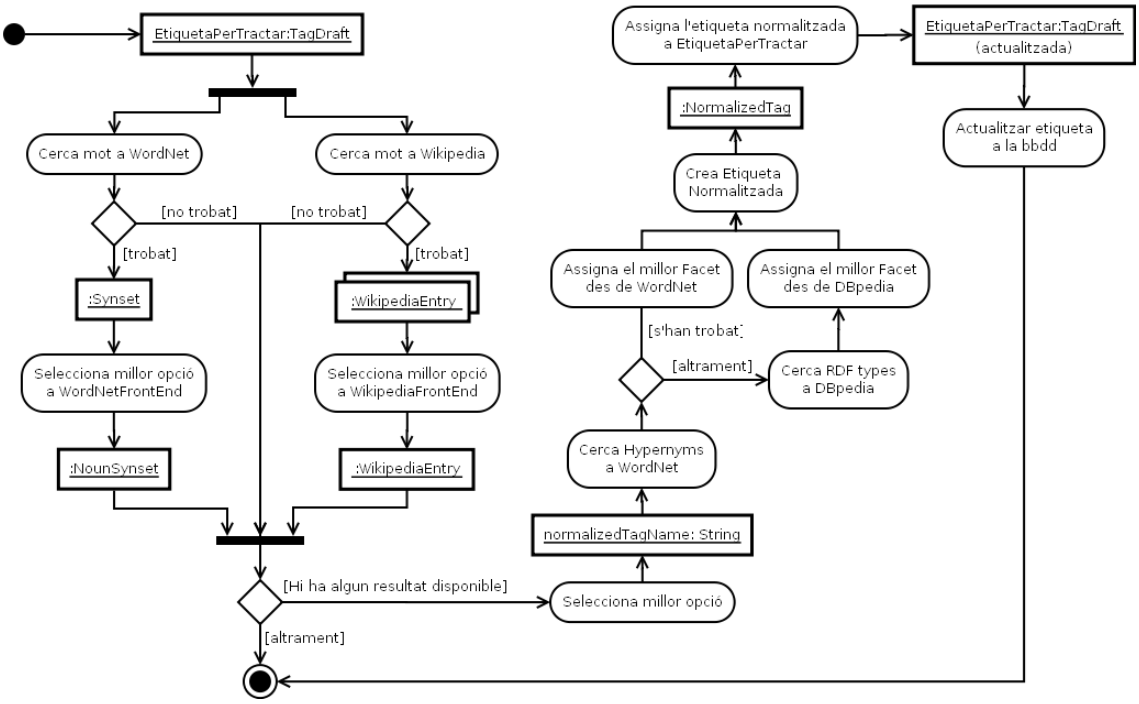


Figura 28 – Diagrama d'activitat del mètode de normalització

Aquest procediment es basa en el treball de Sort Borort de 2009 (veure detalls en la bibliografia). Els canvis respecte al treball de referència es ressalten en cursiva i més endavant es justifiquen adientment.

Com es pot observar, el mot a normalitzar *se cerca tant a la Wikipèdia com a WordNet*. Dels resultats obtinguts *no se selecciona el primer*, sinó que es consideren tots els resultats i se selecciona el millor, més endavant es descriu com es du a terme aquest procediment. Si el mot no s'ha trobat ni a WordNet ni a Wikipedia, l'etiqueta es descarta, altrament el millor resultat de la Wikipèdia i el millor de WordNet *es comparen* amb el mot original *i se'ls assigna una puntuació* que permetrà seleccionar la millor opció, més endavant es descriu com es du a terme aquesta assignació i tria.

L'obtenció de les millors opcions de WordNet i Wikipedia es delega a les classes WordNetFrontEnd i WikipediaFrontEnd respectivament.

Un cop obtingut el nom per a l'etiqueta normalitzada cal assignar-li un Facet. Aquesta funcionalitat es delega a la classe FacetPredictor.

6.7.3 Classe FacetPredictor

Facet predictor rep el nom a assignar a l'etiqueta a normalitzar i suggereix un Facet de la següent manera:

Primer se cerca el mot normalitzat a WordNet i es miren els hypernyms (jerarquia de categories a les quals pertany el mot). Si entre els hypernyms apareix determinats mots s'assignen determinats facets.

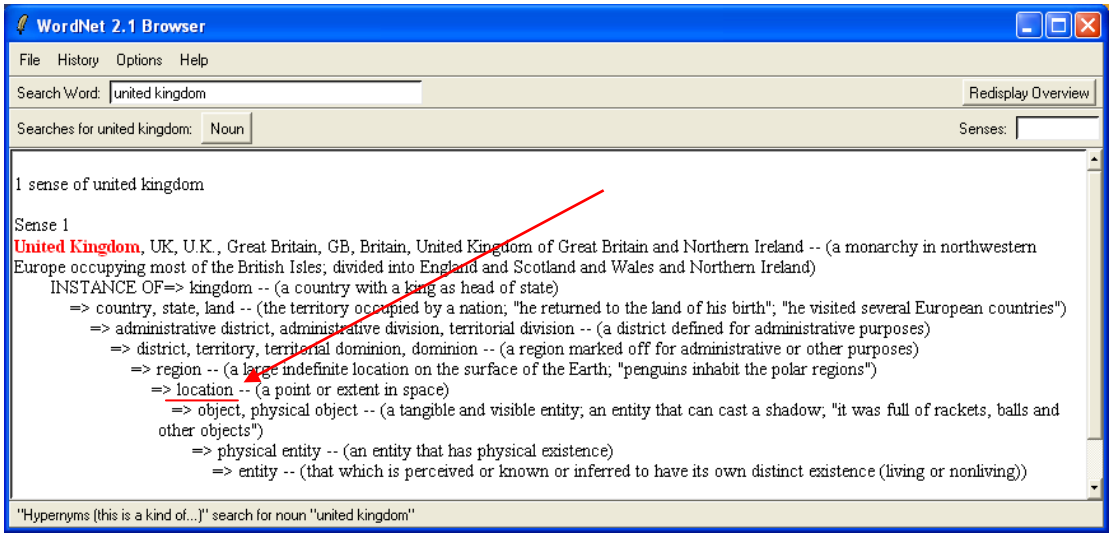


Figura 29 – Jerarquia d'hypernyms del mot *United Kingdom*

Assignació del Facet segons l'hypernym identificat	
Hypernym	Facet
Person	Person
Organization, organisation	Organization
Location, place	Geographic location
Time period	Time related

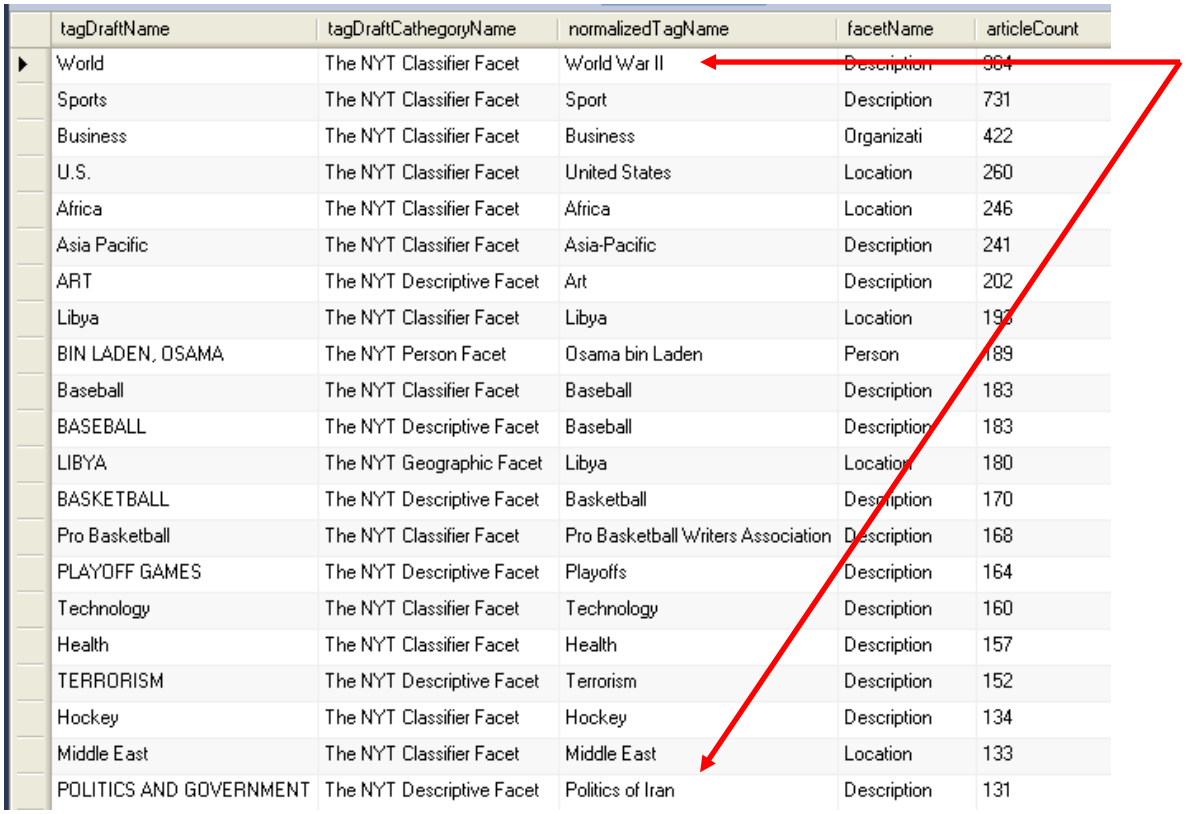
Figura 30 – Taula resum d'assignació de facets des de WordNet

Si el mot no es troba a WordNet o si no té jerarquia d'hypernyms, aleshores se cerca a la DBpedia i s'analitzen els camps rdf:type, que contenen la categorització del mot. Arribats a aquest punt, s'assigna un facet seguint el mateix criteri que a WordNet i si no se n'ha pogut assignar cap aleshores s'assigna el facet *Descriptive Facet*.

Figura 31 – Valors del camp rdf :type per a Paul McCartney a la DBpedia

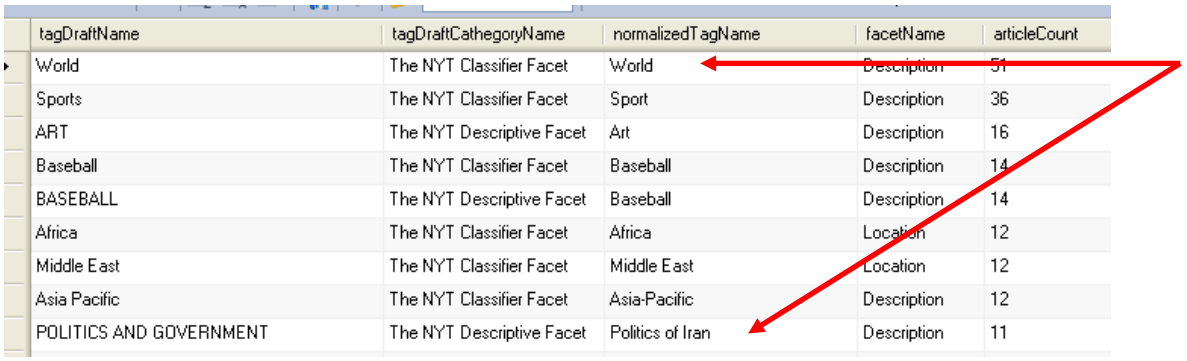
Un cop normalitzada l'etiqueta s'assigna l'etiqueta normalitzada a l'etiqueta original i es desa aquesta interrelació a la base de dades.

Mitjançant aquestes millores s'eviten classificacions errònies de l'estratègia original com la del mot World, que quedava classificat com a World War II.



tagDraftName	tagDraftCathegoryName	normalizedTagName	facetName	articleCount
World	The NYT Classifier Facet	World War II	Description	384
Sports	The NYT Classifier Facet	Sport	Description	731
Business	The NYT Classifier Facet	Business	Organizati	422
U.S.	The NYT Classifier Facet	United States	Location	260
Africa	The NYT Classifier Facet	Africa	Location	246
Asia Pacific	The NYT Classifier Facet	Asia-Pacific	Description	241
ART	The NYT Descriptive Facet	Art	Description	202
Libya	The NYT Classifier Facet	Libya	Location	193
BIN LADEN, OSAMA	The NYT Person Facet	Osama bin Laden	Person	189
Baseball	The NYT Classifier Facet	Baseball	Description	183
BASEBALL	The NYT Descriptive Facet	Baseball	Description	183
LIBYA	The NYT Geographic Facet	Libya	Location	180
BASKETBALL	The NYT Descriptive Facet	Basketball	Description	170
Pro Basketball	The NYT Classifier Facet	Pro Basketball Writers Association	Description	168
PLAYOFF GAMES	The NYT Descriptive Facet	Playoffs	Description	164
Technology	The NYT Classifier Facet	Technology	Description	160
Health	The NYT Classifier Facet	Health	Description	157
TERRORISM	The NYT Descriptive Facet	Terrorism	Description	152
Hockey	The NYT Classifier Facet	Hockey	Description	134
Middle East	The NYT Classifier Facet	Middle East	Location	133
POLITICS AND GOVERNMENT	The NYT Descriptive Facet	Politics of Iran	Description	131

Figura 32 – Resultats amb l'algorisme de normalització de referència



tagDraftName	tagDraftCathegoryName	normalizedTagName	facetName	articleCount
World	The NYT Classifier Facet	World	Description	51
Sports	The NYT Classifier Facet	Sport	Description	36
ART	The NYT Descriptive Facet	Art	Description	16
Baseball	The NYT Classifier Facet	Baseball	Description	14
BASEBALL	The NYT Descriptive Facet	Baseball	Description	14
Africa	The NYT Classifier Facet	Africa	Location	12
Middle East	The NYT Classifier Facet	Middle East	Location	12
Asia Pacific	The NYT Classifier Facet	Asia-Pacific	Description	12
POLITICS AND GOVERNMENT	The NYT Descriptive Facet	Politics of Iran	Description	11

Figura 33 – Resultats amb l'algorisme de normalització modificat*

(*) La diferència d'entrades entre les taules de les dues figures anteriors es déu a que la carrega d'articles en el segon cas era menor.

6.7.4 Classe StringComparator

Els resultats de la Wikipèdia i el de WordNet es comparen amb el mot original i se'ls assigna una puntuació que només pot tenir tres valors: valor màxim en el cas que el mot normalitzat coincideixi totalment amb l'original amb majúscules i minúscules, valor intermedi si coincideix però no amb majúscules i minúscules, i valor mínim en

qualsevol altre cas. Es tria sempre el mot normalitzat de major puntuació i en cas d'empat es tria el mot normalitzat a través de WordNet, ja que en principi WordNet està lliure d'ambigüitats.

6.7.5 Accés a recursos (subpackage *frontEnds*)

En el package *frontEnds* s'inclouen totes les classes que permeten l'accés a recursos externs a l'aplicació, que són Wikipedia, WordNet i DBpedia. Per a cadascun d'aquests recursos hi ha un package específic dins d'aquest. Aquests subpackages són els que es descriuen tot seguit.

6.7.6 Accés a la Wikipedia (subpackage *frontEnds.wikipedia*)

El subpackage *frontEnds.wikipedia* és l'encarregat d'accedir a l'API de la Wikipedia. Disposa de la classe auxiliar WikipediaEntry per a representar cadascun dels resultats obtinguts en la consulta a l'API.

La consulta la du a terme la classe WikipediaFrontEnd i ho fa mitjançant la següent sol·licitud HTTP get:

```
http://en.wikipedia.org/w/api.php?action=opensearch&search=NOM_ETIQUETA&format
                                    =xml
```

On NOM_ETIQUETA és el nom de l'etiqueta que es vol normalitzar.

El format de la resposta es pot indicar mitjançant el camp format, en el cas d'aquest treball XML. Aquest resultat és parsejat i convertit en objectes de la classe WikipediaEntry mitjançant la llibreria *org.w3c.dom*.

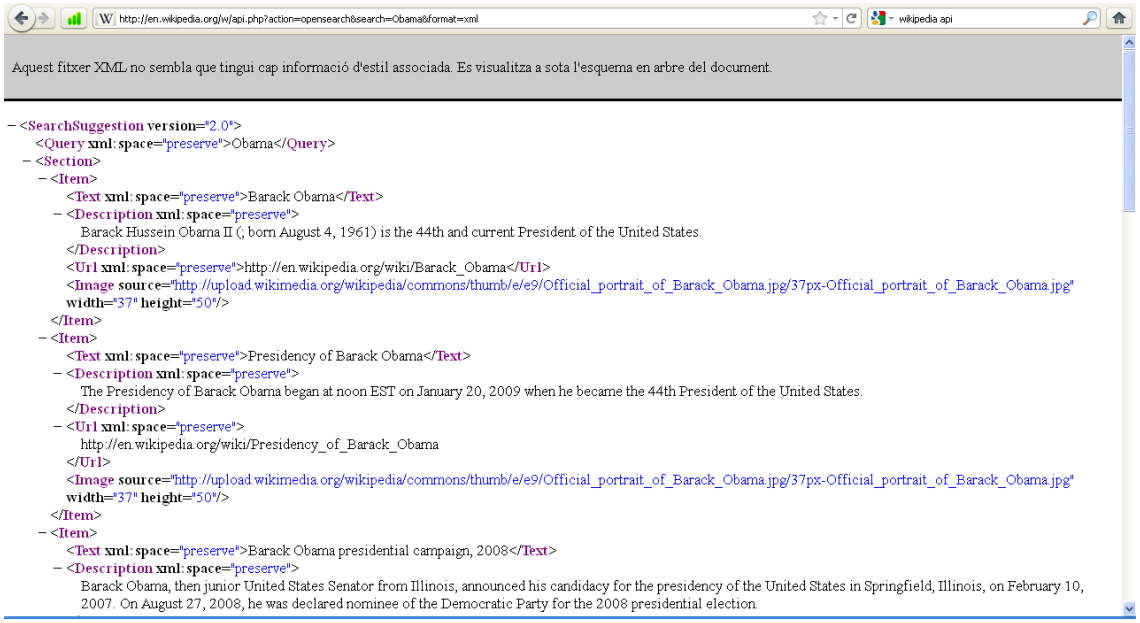


Figura 34 – Resposta de l'API de Wikipedia al mot 'Obama'

Entre les diferents opcions que es retornen de l'API, mitjançant la classe auxiliar StringComparator, s'escull la més similar a l'etiqueta original seguint el criteri per defecte descrit anteriorment.

Més informació sobre l'API de la Wikipedia a:

http://en.wikipedia.org/w/api.php

6.7.7 Accés a WordNet (subpackage *frontEnds.wordnet*)

L'accés a WordNet es du a terme mitjançant la llibreria JAWS. Aquesta llibreria proporciona un conjunt de classes que representen les entrades al diccionari WordNet. A tall de resum, les classes més importants de JAWS són les següents:

- Synset: representa un conjunt de paraules o frases que tenen totes elles un mateix significat (conjunt de sinònims o **syn**onyms **set**). Aquesta classe té un mètode anomenat:

  ```
  getWordForms():String[]
  ```

  que retorna el llistat de lemmes (paraules o frases) que representen a aquest significat. Per exemple, per al primer significat del mot 'world', el llistat de lemmes és:

  ```
  universe, existence, creation, world, cosmos, macrocosm
  ```

  Com es pot observar, el lemma cercat no té per què ser el primer del llistat de lemmes sinònims.

  Després de fer una mica de recerca no s'ha aconseguit esbrinar en quin ordre apareixen els lemmes, però sí que sembla que el primer representa la forma canònica del Synset, la forma més habitual en què es troba representat aquest significat. Això serà important en un futur per a la tria del millor significat.

- SynsetType: representa la categoria sintàctica atorgada a un synset: nom, adverbi, verb, adjectiu...

- NounSynset: tipus concret de synset que es correspon a la categoria sintàctica de *nom* (noun).

  Els NounSynset tenen una particularitat molt interessant, i és que es troben organitzats jeràrquicament, de tal manera que uns són derivats d'uns altres, per exemple: United Nations → world organization → alliance → organization → ...

  La calsse NounSynset disposa dels mètodes getInstanceHypernyms() i getHypernyms(), que retornen el conjunt d'antecessors directes del Synset. Aquest serà el fonament que permetrà anar escalant la jerarquia de hypernyms a la recerca d'una categoria per al Synset.

- WordNetDatabase: és la classe que permet consultar WordNet. Disposa del mètode:

  ```
  getSynsets(String wordForm):Synset[]
  ```

  Aquest mètode retorna el llistat de possibles Synsets per a la paraula cercada. Cadascun dels Synsets representa una de les possibles accepcions o significats de la paraula. Els Synsets venen ordenats per freqüència d'aparició en els textos de prova dels desenvolupadors de WordNet.

El package *frontEnds.WordNet* disposa de la classe WordNetFrontEnd que té delegades dues tasques fonamentals:

- *searchNoun*(String noun): String

  Aquest mètode és el que permet assignar una forma normal al mot *noun*. Per dur a terme aquest procés se cerca el mot a WordNet i se n'extreu els NounSynsets relacionats. Cal recordar que els Synsets venen ordenats per la seva freqüència d'aparició, i que cada Synset té un nom canònic que habitualment representa la forma majoritària en què s'expressa aquest concepte. Així doncs, per triar el millor significat i forma canònica del mot, en absència d'informació addicional, el que es fa és escollir el primer significat que tingui el mot cercat com a forma canònica. Si el mot cercat no és forma canònica de cap significat, aleshores s'escull el primer de tots els significats.

  Per exemple, si se cerca el mot 'world' s'obtenen 8 significats com a nom, on les dues primeres entrades com a mot són:

  ```
  1. universe, existence, creation, world, cosmos, macrocosm --
  (everything that exists anywhere; "they study the evolution
  of the universe"; "the biggest tree in existence")
  2. world, reality -- (all of your experiences that determine
  how things appear to you; "his world was shattered"; "we live
  in different worlds"; "for them demons were as much a part of
  reality as trees were")
  ```

  Seguint els criteris descrits s'escolliria el segon significat com a forma normal del mot.

  Exemples de normalització a WordNet mitjançant aquest procediment:

  ```
  Search word in WordNet:
          Search word 'World': world
          Search word 'NY': New York
          Search word 'Obama': null
  ```

- *suggestFacet*(String noun): Facet

  Aquest mètode és el que permet assignar un Facet a un determinat mot. Seguint el mateix criteri que en el mètode anterior, se selecciona el millor significat per al mot *noun* i s'escala la jerarquia d'hypernyms a la recerca d'una categoria equivalent a

algun dels facets que utilitza l'aplicació. Aquest procés s'aturaria en trobar algun dels mots person, organization, organisation, location, place o time period.

Exemple:

```
Facets suggested by WordNet:
        Searching in WordNet for word 'Socrates':
                Socrates
                philosopher
                scholar
                intellectual
                person
        Facet suggested for 'Socrates': PERSON
        Searching in WordNet for word 'World':
                world
                experience
                content
                cognition
                psychological feature
                abstraction
                abstract entity
                entity
        Facet suggested for 'World': null
        Searching in WordNet for word 'Middle age':
                middle age
                time of life
                time period
        Facet suggested for 'Middle age': TIME_RELATED
        Searching in WordNet for word 'United Nations':
                United Nations
                world organization
                alliance
                organization
        Facet suggested for 'United Nations': ORGANIZATION
        Searching in WordNet for word 'Apple':
                apple
                edible fruit
                produce
                food
                solid
                substance
                physical entity
                entity
        Facet suggested for 'Apple': null
```

Aquelles etiquetes que no hagin pogut ésser classificades d'aquesta manera (Facet = null), passaran a intentar ésser classificades pel frontEnd de la DBpedia. Si DBpedia tampoc és capaç d'assignar-los una categoria, aleshores es classificaran com a Descriptive Subject (exemples de mots no categoritzats per WordNet: World i Apple).

6.7.8 Accés a DBpedia (subpackage *dbpediaFrontEnd*)

El subpackage *frontEnds.dbpedia* és l'encarregat d'accedir a la DBpedia a través d'un punt d'accés públic SPARQL accessible via HTTP.

Com ja s'ha comentat amb anteriroritat, l'accés a la DBpedia s'utilitza per tal de classificicar les etiquetes en un Facet determinat mitjançant l'obtenció de les categories rdf:type assignades als objectes a través de DBpedia.

La consulta la du a terme la classe DBpediaFrontEnd i ho fa mitjançant la següent sol·licitud HTTP get:

```
http://dbpedia.org/sparql?query=
    PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
    PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
    SELECT DISTINCT ?Concept ?Label
    WHERE {
        <http://dbpedia.org/resource/RESOURCE_NAME> rdf:type ?Concept
        .
        ?Concept rdfs:label ?Label
        FILTER ( LANG ( ?Label) = \"en\" || LANG ( ?Label) = "")
}
&format=application/sparql-results xml
```

On RESOURCE_NAME és el nom de l'etiqueta que es vol classificar.

El format de la resposta es pot indicar mitjançant el camp format, en el cas d'aquest treball XML. Aquest resultat és parsejat mitjançant la llibreria *org.w3c.dom* i posteriorment analitzat per extreure'n la categoria.

SPARQL és un llenguatge declaratiu que s'inspira en la lògica proposicional. Es basa en la definició de triplets <valor, camp, etiqueta>, separats per . (operador AND). L'opció FILTER permet filtrar el resultat obtingut de forma senzilla.

Per a més informació sobre la sintaxis SPARQL:

http://www.w3.org/TR/rdf-sparql-query/

Per escollir el facet, en absència de més dades que el nom de l'etiqueta, el que es fa és analitzar els resultats obtinguts de la consulta, que serien anàlegs a la jerarquia d'hypernyms que s'obté a través de WordNet. Si entre aquests mots apareixen els representatius de persona, lloc o ubicació geogràfica, organització o temps, s'assigna el facet corresponent, altrament, no s'assigna facet.

Exemple:

```
Facets suggested by DBpedia:

Suggest Facet for 'Socrates':
[...]
      Thing
      philosopher
      person
        Person FOUND!!!
      Facet suggested for 'Socrates': PERSON
Suggest Facet for 'New York':
[...]
      Thing
      place
        Place FOUND!!!
      Facet suggested for 'New York': GEOGRAPHIC_LOCATION
```

```
Suggest Facet for 'Middle age':
[...]
Facet suggested for 'Middle age': null
      Suggest Facet for 'United Nations':
[...]
      Thing
      Organizations based in New York City
      organisation
        Organization FOUND!!!
      Facet suggested for 'United Nations': ORGANIZATION
Suggest Facet for 'Apple':
      Thing
      plant
      species
      eukaryote
      flowering plant
      Facet suggested for 'Apple': null
```

Les etiquetes que no han obtingut facet, com que tampoc l'han obtingut a l'anàlisi de WordNet, rebran el facet DescriptiveSubject en tornar a cap a la classe FacetPredictor.

6.7.9 Interfície Gràfica d'Usuari (subpackage *gui*)

Aquest subpackage es troba en un estat de desenvolupament molt inicial. Conté una única classe anomenada AnalyzerGUI en la qual únicament es pot configurar cada quants minuts es vol realitzar un anàlisi. El seu funcionament és anàleg a la GUI del package *retrievers*. La classe de control és AnalyzerModule.

6.7.10 Aspectes que han quedat pendents

- Quant a l'ús de la categoria per eliminar ambigüitats

  Sovint les etiquetes venen classificades en categories. Per exemple, Leeds (Sport) o Leeds (UK); en el primer cas fent referència a un equip de baseball i en el segon, a una ciutat britànica.

  Es va desenvolupar un mètode per tal de millorar la selecció del millor candidat entre les diferents accepcions que oferien tant Wikipedia com WordNet tenint en compte la categoria com venien classificades les etiquetes.

  Aquest mètode consistia en fer un recorregut en amplada dels diferents hypernyms (de més específic a menys) de les diferents accepcions del mot en qüestió per a cadascun dels hypernyms del millor candidat per a la categoria, tot comparant els hypernyms d'un mot amb els de l'altre. Els hypernyms de la categoria s'han de recórrer ordenadament de més específic a menys. Fet així, la condició d'acabament és trobar coincidència entre termes. Quan es troba aquesta coincidència se sumen les profunditats dels hypernyms i això pot servir d'estimació de la distància semàntica entre els dos mots. A menor distància, major similitud.

  En el cas dels mots que no tenen entrada a WordNet, es va desenvolupar un mètode alternatiu per extreure'n els hypernyms des de la DBpedia, on aquestos no venen

ordenats jeràrquicament i, per tant, la mesura de distància ja no és vàlida. Aquest mètode consistia en anar analitzant cadascun dels hypernyms obtinguts i identificar-ne relacions jeràrquiques per tal d'ordenar-los.

Tot i que l'aproximació inicial pot semblar bona, el resultat no ho va estar. Molt poques vegades el resultat final oferia una millora respecte al mètode original i el temps de càlcul era considerablement major, sobretot si es tenien en compte les entrades de la DBpedia, ja que s'havien de fer moltes consultes HTTP.

La causa d'aquest fracàs es fa evident amb el següent exemple:

> Leeds, com a equip de futbol, és una organització o grup social, mentre que sport és una activitat. En la seva jerarquia d'hypernyms tots dos conceptes segueixen camins diferents i no coincideixen fins a ésser considerat tots dos com a entitats, però a aquest nivell, tots dos conceptes han perdut ja tota la seva essència.
>
> El mateix passaria, per exemple, amb Bacon en la categoria Philosophy, ja que Bacon es tracta d'un filòsof i per tant una entitat física, mentre que Philosophy seria una entitat abstracta.
>
> I així amb molts casos més.

Per tant, queda pendent la recerca o elaboració d'algun mètode que utilitzi la potència de tenir els conceptes prèviament categoritzats per tal de seleccionar el millor candidat durant el procediment de normalització.

- Millorar el procediment de tria del millor candidat mitjançant l'addició de mètriques de comparació entre paraules i/o frases.

  Per millorar en la tria de la millor etiqueta normalitzada també es pot provar d'utilitzar algun sistema de puntuació basat en alguna mètrica entre mots. En aquest sentit, la llibreria opensource Java Text Mining Toolkit ofereix múltiples possibilitats: http://jtmt.sourceforge.net/

6.8 Mòdul Visualitzador de resultats (package *resultsExplorer*)

Aquest mòdul és el que permet visualitzar els resultats de l'anàlisi. Les classes pertinents a aquest mòdul es troben en el package *resultsExplorer* i es troben estructurades de la següent manera:

6.8.1 Classe ResultsExplorerModule

És la classe principal del mòdul. És la controladora que rep les instruccions de la interfície gràfica d'usuari (GUI), executa les comandes pertinents i retorna les dades a mostrar la GUI.

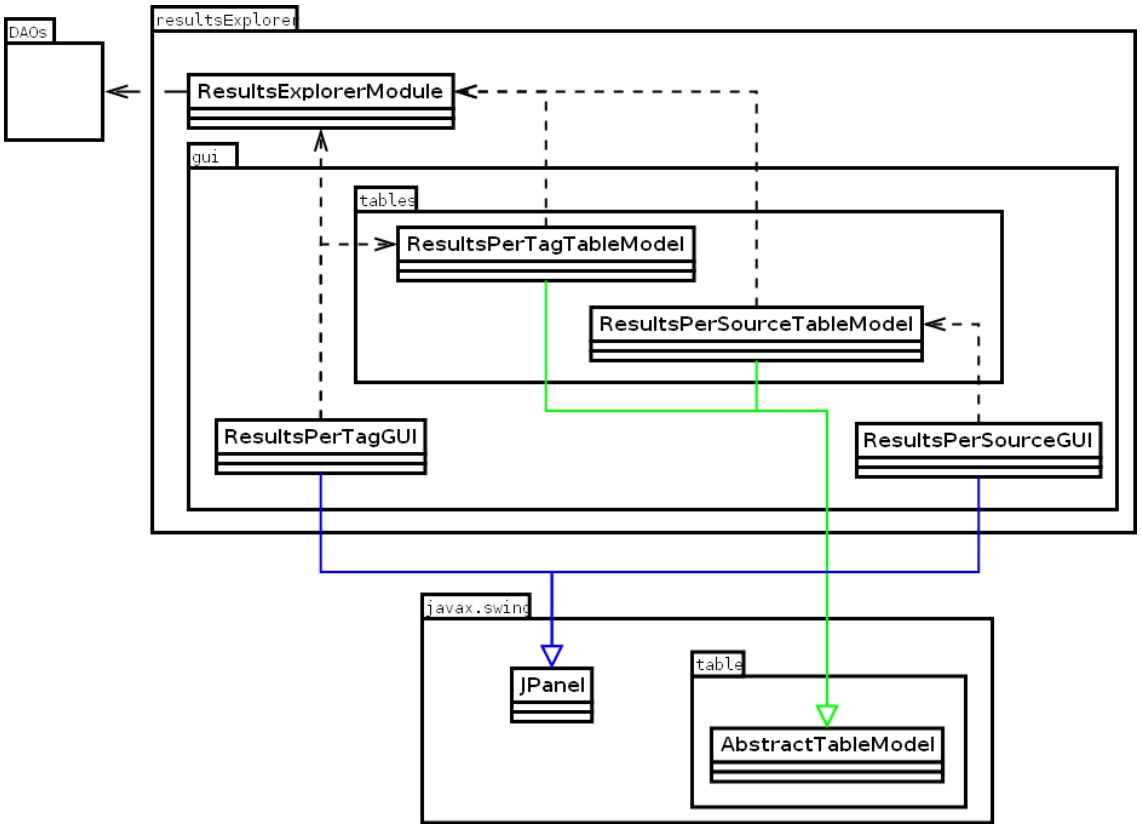


Figura 35 – Diagrama UML del mòdul explorador de resultats (resultsExplorer)

6.8.2 Vistes dels resultats (subpackage *gui*)

Conté les pantalles *ResultsPerTagGUI* i *ResultsPerSourceGUI*, que permeten visualitzar els resultats de l'anàlisi en forma d'etiquetes normalitzades i la seva freqüència per font, i etiquetes originals i etiqueta normalitzada assignada, respectivament.

Les dues pantalles hereten de JPanel i són carregades sota demanda com a subpantalles en la pantalla principal de l'aplicació, que empra un CardLayout per gestionar el canvi de subpantalla.

Els resultats es mostren de forma tabular dins les pantalles utilitzant objectes de tipus *javax.swing.JList*, tot carregant les taules amb els objectes de tipus TableModel que es descriuen en el subpackage *gui.tables*.

6.8.3 Representació de les dades (subpackage *gui.tables*)

Aquest package conté els models de dades que es carreguen en les JList de les pantalles del package *gui*. Aquests models hereten de la classe *javax.swing.table.AbstractTableModel* i són les dues classes següents que contenen els resultats de les consultes SQL que es detallen a continuació:

- ResultsPerSourceTableModel

  Conté les dades i percentatges d'etiquetes normalitzades per font de notícies i en global.

  Les consultes SQL emprades per obtenir aquestes dades han estat les següents:

  - Per a l'obtenció dels percentatges globals:

    Per obtenir la quantitat global d'articles on apareix cada etiqueta:

    ```
    SELECT
      NormalizedTag.normalizedTagId,
      normalizedTagName,
      normalizedTagFacetId,
      COUNT(articleId) AS count
    FROM
      NormalizedTag NATURAL JOIN ArticleHasNormalizedTag
    GROUP BY NormalizedTag.normalizedTagId
    ```

    Per obtenir el percentatge cal ara dividir els resultats obtinguts en la consulta anterior pel total d'articles existents:

  ```
  SELECT COUNT(articleId) AS count FROM Article
  ```

  - Per a l'obtenció dels percentatges de cada etiqueta per a cada font:

    Primer cal obtenir la quantitat d'articles d'aquella font on apareix aquesta etiqueta:

    ```
    SELECT COUNT(articleId) AS freq
    FROM ArticleHasNormalizedTag NATURAL JOIN Article
    WHERE normalizedTagId = ? AND articleSourceId = ?
    ```

    Després cal dividir el resultat pel total d'articles existents per a aquella font:

    ```
    SELECT COUNT(articleId)
    FROM Article
    WHERE articleSourceId = ?
    ```

- ResultsPerTagTableModel

  Conté les dades de totes les etiquetes entrades al sistema i la seva normalització i classificació en facets.

  La consulta SQL emprada per obtenir aquestes dades ha estat la següent:

  ```
  SELECT
    TagDraft.tagDraftId AS id,
    TagDraft.tagDraftName AS name,
    Cathegory.tagDraftName AS cathegory,
    TagDraft.tagDraftFacetId AS facet,
    normalizedTagName AS normalizedName,
    normalizedTagFacetId AS predictedFacet
  ```

```
FROM
  (TagDraft LEFT OUTER JOIN TagDraft AS Cathegory ON
TagDraft.tagDraftCathegoryId = Cathegory.tagDraftId)
  LEFT OUTER JOIN NormalizedTag ON TagDraft.normalizedTagId =
NormalizedTag.normalizedTagId
```

6.8.4 Vista d'avaluació orientativa dels resultats obtinguts

En la pantalla *ResultsPerTagGUI* apareixen dos camps que ens permeten fer una estimació de la potència del mètode de normalització. Aquests camps són els següents:

- Amount of tags normalized (%)

  Percentatge d'etiquetes originals que han pogut ésser vinculades a una etiqueta normalitzada.

- Accuracy based on facet prediction (%)

  Percentatge d'etiquetes normalitzades per a les quals la seva classificació en facets es correspon a la de l'etiqueta original.

  Donat que en el NYT no s'etiqueta per al facet de tipus TIME_RELATED, no es consideraran aquestes etiquetes en el càlcul.

6.8.5 Aspectes que han quedat pendents

Caldria revisar a fons aquesta GUI per tal de mostrar les dades de forma més clara a l'usuari final i més d'acord amb els objectius del projecte, ja que ara l'usuari ha de fer un esforç per entendre els resultats i aquesta tasca li hauria de venir facilitada.

6.9 Pantalla principal i menú de l'aplicació (package *gui*)

En l'arrel del package principal del programa (*newsAnalyzer*) s'hi troba el package *gui*. Aquest package conté una única classe NewsAnalyzerGUI, que conté el menú principal de l'aplicació i s'encarrega d'instanciar aquelles pantalles o mòduls que s'hagin de carregar.

6.10 Lliurable final

El lliurable final de l'aplicació consta del directori newsAnalyzer amb els següents elements:

- directori *src*

  Conté el codi font del programa.

- Directori *sql*

  Conté els scripts de creació de la base de dades.

- Directori *externalLibs*

  Conté totes les llibreries externes que es requereixen per tal de compilar i executar el programa.

- Fitxer *compile.bat*

  Conté els scripts de compilació del programa.

En compilar el programa mitjançant el script compile.bat es genera el directori dist, que conté tots els elements necessaris per a la distribució i execució del programa (a excepció del fitxers d'instal·lació de WordNet i del servidor de MySQL, que s'han d'instal·lar a part).

En la següent secció d'aquesta memòria es descriu amb tot detall com compilar, instal·lar i executar el programa tot proporcionant els manuals adients per a cadascun d'aquests procediments.

7 Proves i discussió de resultats

Tot seguit es descriuran les diferents proves dutes a terme durant el desenvolupament de la versió inicial d'aquesta aplicació i es durà a terme un breu exercici de reflexió i discussió dels resultats obtinguts.

7.1 Proves en el procediment de recol·lecció de notícies

7.1.1 Volum de les dades

Un dels punts crítics en aquest apartat era el gran creixement de la base de dades.

Sobre una mostra d'articles que va del 15/04/2011 fins al 11/06/2011, la següent consulta SQL mostra la quantitat promig d'articles recollits al dia per font:

```
SELECT articleSourceName, AVG(t.n)
FROM
    (SELECT articleSourceId, articleDate, COUNT(articleId) AS n
        FROM article
        GROUP BY articleSourceId, articleDate
    ) AS t
    NATURAL JOIN
    ArticleSource
GROUP BY t.articleSourceId;
```

articleSourceName	AVG(t.n)
The Guardian	116.3793
The New York Times	159.1552

Figura 36 – Quantitat promig d'articles per font i dia

Això suposa un total de **15981** articles en poc menys de 2 mesos.

La quantitat d'assignacions d'etiquetes a articles per dia i font en aquest període fou la següent:

```
SELECT articleSourceName, AVG(t.n)
FROM
    (SELECT articleSourceId, articleDate, COUNT(tagDraftId) AS n
        FROM article NATURAL JOIN articleHasTagDraft
        GROUP BY articleSourceId, articleDate
    ) AS t
    NATURAL JOIN
    ArticleSource
GROUP BY t.articleSourceId;
```

articleSourceName	AVG(t.n)
The Guardian	962.9138
The New York Times	987.2931

Figura 37 – Quantitat promig d'assignacions d'etiquetes per font i dia

Això suposa **113112** assignacions en aquest període.

Si cada cop que es recull una assignació es guardés l'etiqueta com una nova etiqueta a la base de dades, la quantitat d'etiquetes a guardar seria també 113112.

Per tal d'incrementar l'eficiència de la recol·lecció es va crear el mètode **load** en els DAOs de les diferents classes amb què treballa l'aplicació. Aquest mètode cerca primer si l'objecte es troba a la base de dades i si es troba el carrega per a l'aplicació per tal de fer-ne ús.

Aplicant el mètode load a les etiquetes, no es crea de nou l'etiqueta cada vegada sinó que es referencia aquella etiqueta ja existent. Això diverses conseqüències:

- Estalvi d'espai: la quantitat d'etiquetes desades en el període de temps anterior va ser de **9094**, una mica menys d'un 10%. Això suposa estalviar pràcticament la meitat de l'espai.
- Eliminació de duplicats en la base de dades
- Estalvi de temps posteriorment en el procediment de normalització, ja que el resultat de la normalització es pot desar associat a l'etiqueta original.

7.1.2 El factor temps en la recol·lecció

Un altre punt crític és el temps que triga la recol·lecció.

El temps aproximat de recol·lecció de les notícies depèn de la connexió a Internet, la velocitat de procés i la d'accés a la base de dades.

- Temps habitual per font de descàrrega i procés de les notícies d'un dia:

  10-12 segons (amb el mode debug activat).

- Temps habitual per font de descàrrega, procés i desat a base de dades de les notícies d'un dia:

  40 segons (amb el mode debug activat).

Dels resultats anteriors es desprèn que l'element limitant és l'accés a la base de dades.

7.1.3 Discussió dels resultats

El procediment té un temps raonable però encara es pot treballar més per millorar el rendiment de l'accés a la base de dades en el desat dels articles.

7.2 Proves en el procediment d'anàlisi de les notícies

7.2.1 Model de referència de Borort, S. (2009)

En el package newsAnalyzer.analyzer.utils s'han implementat les classes ReferenceTagNormalizer i ReferenceFacetPredictor, que implementen els mètodes de normalització i predicció de facets descrits en el treball de Borort, S. (2009).

7.2.1.1 Quant a precisió la normalització

El resultat de la normalització mitjançant aquestes classes deixa força que desitjar, en una mostra de les 20 primeres etiquetes tractades per al dia 11/06/2011 el resultat és el següent:

Etiqueta	Categoria	Etiqueta normalitzada
URBAN AREAS		Urban area
HISTORIC BUILDINGS AND SITES		
ARCHITECTURE		Architecture
RACE AND ETHNICITY		Race and ethnicity in the United States Census
UNITED STATES POLITICS AND GOVERNMENT		
BLACKS		Blacksmith
CRIME AND CRIMINALS		
DRUG ABUSE AND TRAFFIC		
CARBON DIOXIDE		Carbon dioxide
GREENHOUSE GAS EMISSIONS		Greenhouse gas
ENVIRONMENT		Environment
Health		Health
Diseases, Conditions, and Health Topics	Health	
Diet	Diseases, Conditions, and Health Topics	Diet of Japan
CHILDREN AND YOUTH		Sports school
WEIGHT		Weight
DIET AND NUTRITION		
MCDONALD'S CORPORATION		McDonald's
World		World War II
Countries and Territories	World	List of countries and territories by land and maritime borders

Figura 38 – Resultats obtinguts en la normalització amb el mètode de referència

Amb aquesta mostra reduïda d'etiquetes ja es pot observar que hi ha força problemes.

Concretament, un 25% de les etiquetes han quedat sense normalitzar, i de les normalitzades, un 40% s'han normalitzat de forma errònia.

Tot i que no és una mostra prou representativa, ja s'intueix que és un aspecte a millorar.

7.2.1.2 Quant a la classificació per facets

Acte seguit es mostra el resultat per a una mostra d'etiquetes del procediment de referència de classificació per facets.

Etiqueta	Facet	Facet predit
MCDONALD'S CORPORATION	ORGANIZATION	DESCRIPTIVE SUBJECT
JAPAN	GEOGRAPHIC LOCATION	GEOGRAPHIC_LOCATION (By DBpedia)
WASHINGTON	GEOGRAPHIC LOCATION	DESCRIPTIVE_SUBJECT
REPUBLICAN PARTY	ORGANIZATION	DESCRIPTIVE_SUBJECT
YUCCA MOUNTAIN (NEV)	GEOGRAPHIC LOCATION	GEOGRAPHIC LOCATION (By DBpedia)
NUCLEAR REGULATORY COMMISSION	ORGANIZATION	ORGANIZATION (By WordNet)
GEORGIA	GEOGRAPHIC LOCATION	GEOGRAPHIC LOCATION (By DBpedia)
HENRY, THIERRY	PERSON	PERSON (By DBpedia)
NEW YORK YANKEES	ORGANIZATION	DESCRIPTIVE SUBJECT
TURKEY	GEOGRAPHIC LOCATION	GEOGRAPHIC LOCATION (By DBpedia)
MIDDLE AGE	TIME RELATED	TIME RELATED (By WordNet)

Figura 39 – Resultats de la classificació en facets amb el mètode de referència

Com es pot observar, el mètode funciona força bé, algunes vegades essent DBpedia qui determina el facet i altres WordNet. La taxa d'error del mètode dependrà no només de la capacitat de predicció del mètode sinó també de la taxa d'encert en la normalització de l'etiqueta.

7.2.1.3 Quant al rendiment

Allò més bo que tenen els mètodes de referència és la seva rapidesa de càlcul. El temps d'anàlisi dependrà de la velocitat de normalització i de la velocitat de predicció del facet.

Així, en normalitzar una etiqueta es triga, de mitjana, **0,625** segons, mentre que normalitzar i predir els facets suposa, de mitjana, **0,76** segons.

Per tant, com es pot observar, la major part del temps es dedica en la normalització i, en concret, la major part del temps es deu a l'accés HTTP als recursos.

7.2.2 Millores introduïdes en aquest treball

En les seccions anteriors s'han comentat les millores introduïdes en aquest treball respecte al de referència. Tot seguit s'analitzen aquestes millores respecte als resultats.

7.2.2.1 Quant a la precisió en la normalització

El resultat de la normalització mitjançant les classes millorades repercuteix en una lleugera millora en el resultat respecte al model de referència, per a la mateixa mostra de 20 etiquetes emprades en la prova de normalització de les classes de referència, el valor obtingut en les classes millorades ha estat el següent:

Etiqueta	Categoria	Etiqueta normalitzada
URBAN AREAS		Urban area
HISTORIC BUILDINGS AND SITES		
ARCHITECTURE		Architecture
RACE AND ETHNICITY		Race and ethnicity in the United States Census
UNITED STATES POLITICS AND GOVERNMENT		
BLACKS		**Blacks**
CRIME AND CRIMINALS		
DRUG ABUSE AND TRAFFIC		
CARBON DIOXIDE		Carbon dioxide
GREENHOUSE GAS EMISSIONS		Greenhouse gas
ENVIRONMENT		environment
Health		Health
Diseases, Conditions, and Health Topics	Health	
Diet	Diseases, Conditions, and Health Topics	**Diet**
CHILDREN AND YOUTH		Sports school
WEIGHT		Weight
DIET AND NUTRITION		
MCDONALD'S CORPORATION		McDonald's
World		**World**
Countries and Territories	World	List of countries and territories by land and maritime borders

Figura 40 – Resultats de la normalització amb el mètode millorat

Amb aquesta mostra reduïda d'etiquetes, tot i que no és prou representativa, ja es pot observar que hi ha una lleugera millora respecte al mètode de referència.

Concretament, per a aquesta mostra s'ha disminuït en un 50% la quantitat d'etiquetes que s'han normalitzat de forma errònia (s'han ressaltat amb color blau).

7.2.2.2 Quant a la classificació per facets

En aquest cas la millora es deu a la millora en el procediment de normalització, ja que l'únic canvi en el procediment ha estat que en comptes de cercar en primer lloc a la DBpedia, se cerca en primer lloc a WordNet per tal de millorar el temps de resposta, ja que s'estalvia la consulta HTTP. En conseqüència, la millora en el percentatge de facets predits correctament és una mesura aproximada de la millora en el procediment de normalització subjacent.

Com ja s'ha comentat, la millora introduïda en el mètode de normalització és lleugera. En una mostra de 50 etiquetes a l'atzar, la millora en els facets ha arribat a un 20% menys d'errors en la classificació, el que passa és que l'error ja era molt baix, amb la

qual cosa la quantitat d'etiquetes que han canviat de classificació ha estat únicament un 2%.

Aquests valors són força insignificants, el que passa és que la millora obtinguda en la normalització és molt superior, però no s'observa en la predicció de facets, ja que la majoria d'etiquetes pertanyen a la categoria DESCRIPTIVE SUBJECT, tant en la seva forma normal correcta com en la incorrecta. De tota manera, sí que sembla directa la relació millora en la predicció i millora en la normalització i, per tant, seria vàlid, però no quantitatiu, utilitzar-lo per justificar una millora en la normalització.

7.2.2.3 Quant al rendiment

El temps de còmput del mètode millorat difereix lleugerament del de referència:

- Mètode de referència: promig de **0,78** segons per etiqueta (sense debug).
- Mètode millorat: promig de **0,98** segons per etiqueta (sense debug).

Per tant, el mètode millorat suposa un increment d'un **25%** en el temps de còmput.

D'altra banda, els temps calculats fins al moment no inclouen l'accés a la base de dades per recollir les notícies pendents d'analitzar, ni desar el resultat de l'anàlisi, que són els procediments més costosos, juntament amb el fet que hi ha un gran volum d'etiquetes per normalitzar.

Per tal de disminuir els temps s'han provat i acceptat les següents millores:

- Utilitzar una tècnica homòloga a la del procediment de recol·lecció de notícies mitjançant el mètode **load**, que permet carregar les etiquetes i etiquetes normalitzades si ja existeixen o crear-les si no existien abans.
- Utilitzar **taules hash** que continguin totes les etiquetes tractades fins al moment i una totes les etiquetes normalitzades obtingudes fins al moment amb la intenció d'estalviar accessos a la base de dades, que són els més costosos. El cost en memòria d'aquestes taules és assumible donat que la quantitat d'etiquetes noves decreix amb el temps (ja que la probabilitat que una etiqueta ja s'hagi introduït amb anterioritat és més gran). En dos mesos la quantitat d'etiquetes obtingudes ha estat de prop de 10000.

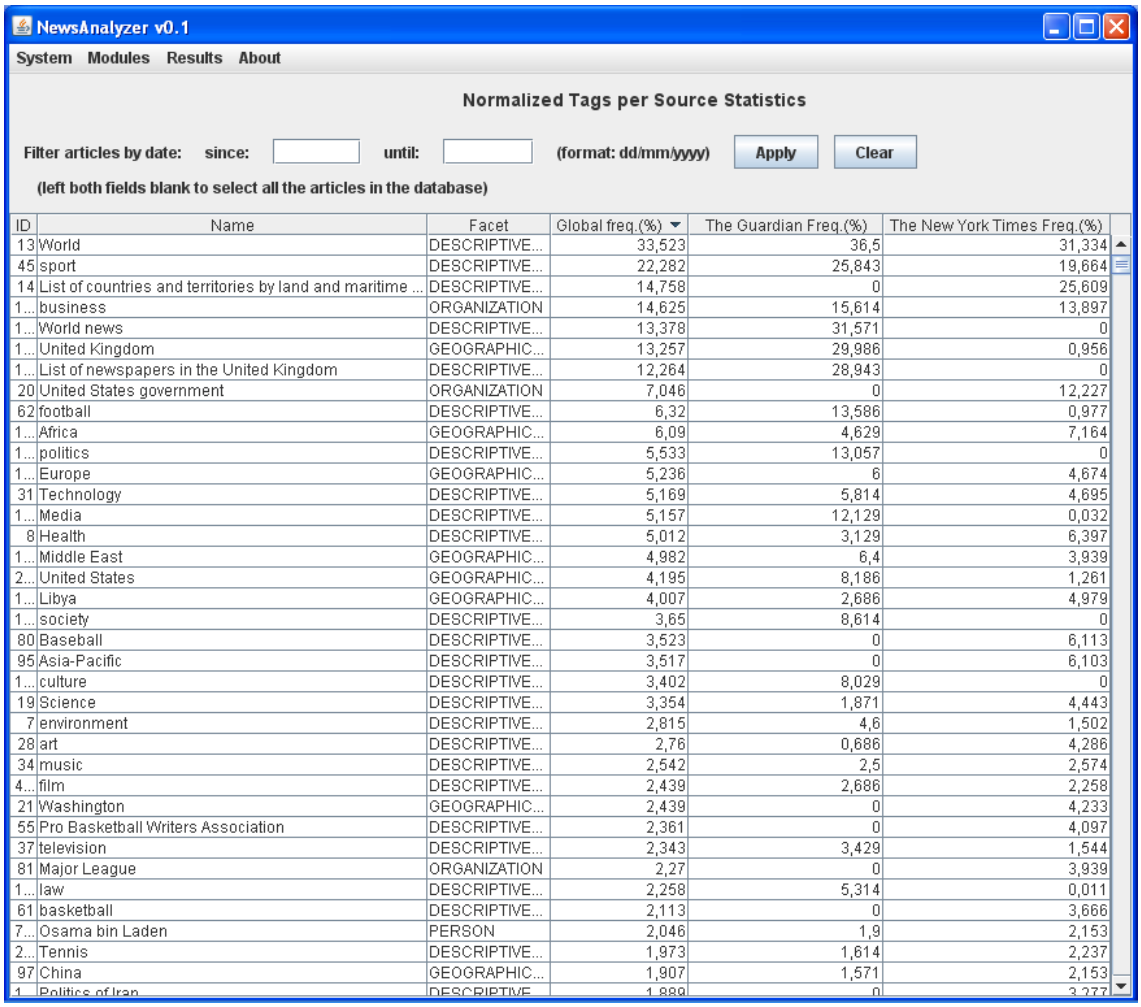
7.2.3 Discussió dels resultats

El procediment millorat té un temps molt raonable, ja que un increment en un 25% del temps de còmput és assumible. Donat que el procediment limitant és l'accés i desat a la base de dades, cal treballar més per millorar-ne el rendiment.

7.3 Proves quant al compliment dels objectius d'aquest treball

7.3.1 Temàtiques globals (entre totes les fonts)

L'objectiu d'aquest treball és poder comparar la temàtica de les publicacions entre fonts. Amb aquesta finalitat s'ha desenvolupat la pantalla ResutlsPerSourceJPanel que, per al període entre el 13/04/2011 i el 11/04/2011 té el següent aspecte:



ID	Name	Facet	Global freq.(%)	The Guardian Freq.(%)	The New York Times Freq.(%)
13	World	DESCRIPTIVE...	33,523	36,5	31,334
45	sport	DESCRIPTIVE...	22,282	25,843	19,664
14	List of countries and territories by land and maritime ...	DESCRIPTIVE...	14,758	0	25,609
1...	business	ORGANIZATION	14,625	15,614	13,897
1...	World news	DESCRIPTIVE...	13,378	31,571	0
1...	United Kingdom	GEOGRAPHIC...	13,257	29,986	0,956
1...	List of newspapers in the United Kingdom	DESCRIPTIVE...	12,264	28,943	0
20	United States government	ORGANIZATION	7,046	0	12,227
62	football	DESCRIPTIVE...	6,32	13,586	0,977
1...	Africa	GEOGRAPHIC...	6,09	4,629	7,164
1...	politics	DESCRIPTIVE...	5,533	13,057	0
1...	Europe	GEOGRAPHIC...	5,236	6	4,674
31	Technology	DESCRIPTIVE...	5,169	5,814	4,695
1...	Media	DESCRIPTIVE...	5,157	12,129	0,032
8	Health	DESCRIPTIVE...	5,012	3,129	6,397
1...	Middle East	GEOGRAPHIC...	4,982	6,4	3,939
2...	United States	GEOGRAPHIC...	4,195	8,186	1,261
1...	Libya	GEOGRAPHIC...	4,007	2,686	4,979
1...	society	DESCRIPTIVE...	3,65	8,614	0
80	Baseball	DESCRIPTIVE...	3,523	0	6,113
95	Asia-Pacific	DESCRIPTIVE...	3,517	0	6,103
1...	culture	DESCRIPTIVE...	3,402	8,029	0
19	Science	DESCRIPTIVE...	3,354	1,871	4,443
7	environment	DESCRIPTIVE...	2,815	4,6	1,502
28	art	DESCRIPTIVE...	2,76	0,686	4,286
34	music	DESCRIPTIVE...	2,542	2,5	2,574
4...	film	DESCRIPTIVE...	2,439	2,686	2,258
21	Washington	GEOGRAPHIC...	2,439	0	4,233
55	Pro Basketball Writers Association	DESCRIPTIVE...	2,361	0	4,097
37	television	DESCRIPTIVE...	2,343	3,429	1,544
81	Major League	ORGANIZATION	2,27	0	3,939
1...	law	DESCRIPTIVE...	2,258	5,314	0,011
61	basketball	DESCRIPTIVE...	2,113	0	3,666
7...	Osama bin Laden	PERSON	2,046	1,9	2,153
2...	Tennis	DESCRIPTIVE...	1,973	1,614	2,237
97	China	GEOGRAPHIC...	1,907	1,571	2,153
1	Politics of Iran	DESCRIPTIVE...	1,889	0	3,277

Figura 41 – Etiquetatge normalitzat global més abundant

Com es pot apreciar, en aquest període es pot afirmar que les notícies més abundants en global han estat les internacionals.

L'ordre en la temàtica, de més a menys abundant: esports, negocis, política, tecnologia, salut, societat, cultura, ciència, medi ambient, art, música, cinema...

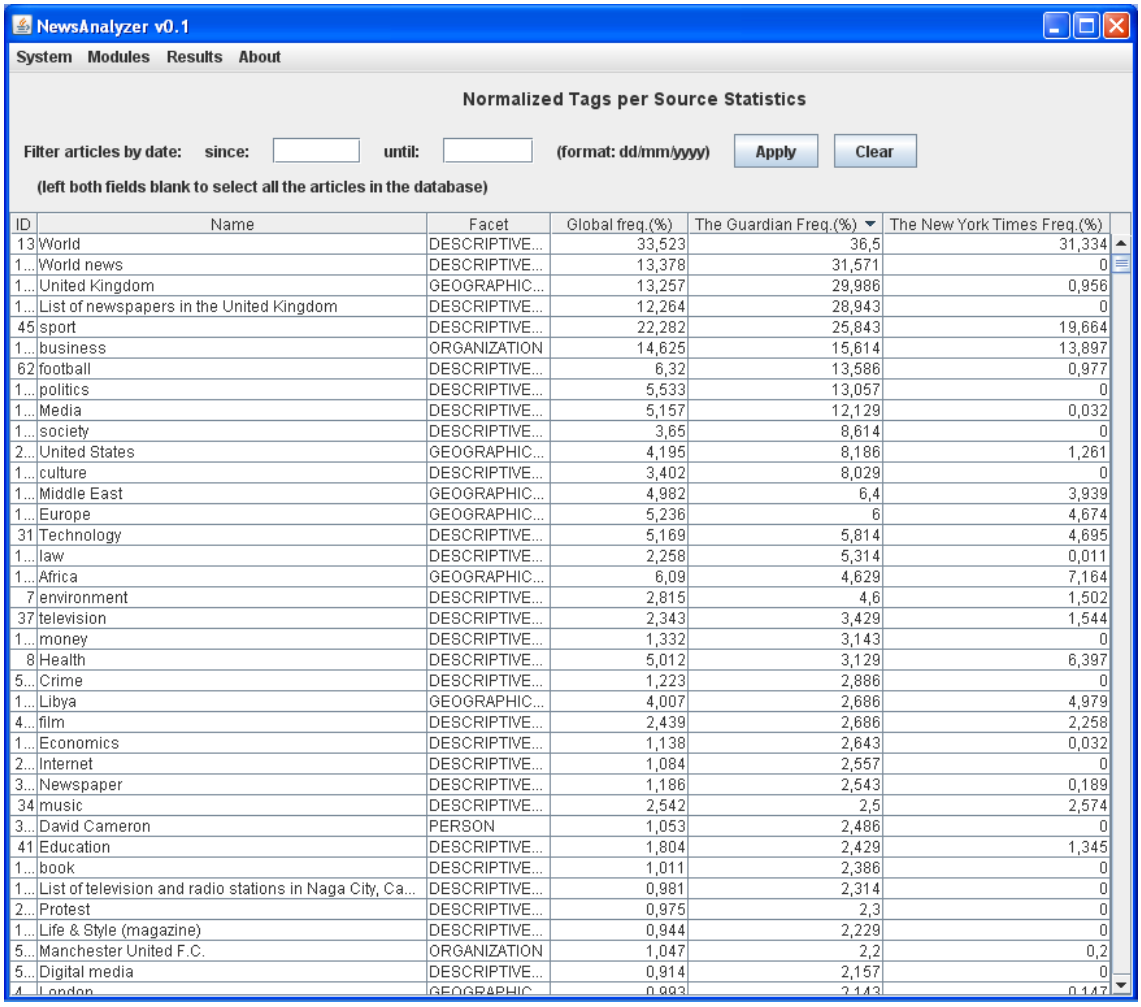
Si prenem les dues temàtiques més importants:

- Entre les notícies internacionals destaquen les referents a Àfrica, i no és gens estrany, donades les revoltes populars que s'estan produint al nord d'aquest continent.

- Entre els esports es pot deduir que les notícies més abundants fan referència al futbol i al beisbol.

7.3.2 Temàtiques més habituals al diari The Guardian

Si ara ens fixem en la freqüència de les etiquetes en el diari The Guardian observem el següent:



ID	Name	Facet	Global freq.(%)	The Guardian Freq.(%)	The New York Times Freq.(%)
13	World	DESCRIPTIVE...	33,523	36,5	31,334
1...	World news	DESCRIPTIVE...	13,378	31,571	0
1...	United Kingdom	GEOGRAPHIC...	13,257	29,986	0,956
1...	List of newspapers in the United Kingdom	DESCRIPTIVE...	12,264	28,943	0
45	sport	DESCRIPTIVE...	22,282	25,843	19,664
1...	business	ORGANIZATION	14,625	15,614	13,897
62	football	DESCRIPTIVE...	6,32	13,586	0,977
1...	politics	DESCRIPTIVE...	5,533	13,057	0
1...	Media	DESCRIPTIVE...	5,157	12,129	0,032
1...	society	DESCRIPTIVE...	3,65	8,614	0
2...	United States	GEOGRAPHIC...	4,195	8,186	1,261
1...	culture	DESCRIPTIVE...	3,402	8,029	0
1...	Middle East	GEOGRAPHIC...	4,982	6,4	3,939
1...	Europe	GEOGRAPHIC...	5,236	6	4,674
31	Technology	DESCRIPTIVE...	5,169	5,814	4,695
1...	law	DESCRIPTIVE...	2,258	5,314	0,011
1...	Africa	GEOGRAPHIC...	6,09	4,629	7,164
7	environment	DESCRIPTIVE...	2,815	4,6	1,502
37	television	DESCRIPTIVE...	2,343	3,429	1,544
1...	money	DESCRIPTIVE...	1,332	3,143	0
8	Health	DESCRIPTIVE...	5,012	3,129	6,397
5...	Crime	DESCRIPTIVE...	1,223	2,886	0
1...	Libya	GEOGRAPHIC...	4,007	2,686	4,979
4...	film	DESCRIPTIVE...	2,439	2,686	2,258
1...	Economics	DESCRIPTIVE...	1,138	2,643	0,032
2...	Internet	DESCRIPTIVE...	1,084	2,557	0
3...	Newspaper	DESCRIPTIVE...	1,186	2,543	0,189
34	music	DESCRIPTIVE...	2,542	2,5	2,574
3...	David Cameron	PERSON	1,053	2,486	0
41	Education	DESCRIPTIVE...	1,804	2,429	1,345
1...	book	DESCRIPTIVE...	1,011	2,386	0
1...	List of television and radio stations in Naga City, Ca...	DESCRIPTIVE...	0,981	2,314	0
2...	Protest	DESCRIPTIVE...	0,975	2,3	0
1...	Life & Style (magazine)	DESCRIPTIVE...	0,944	2,229	0
5...	Manchester United F.C.	ORGANIZATION	1,047	2,2	0,2
5...	Digital media	DESCRIPTIVE...	0,914	2,157	0
4...	London	GEOGRAPHIC...	0,993	2,143	0,147

Figura 42 – Etiquetatge més abundant al diari The Guardian

Es mantenen les notícies internacionals com a les més abundants, seguides de les notícies del Regne Unit, com és natural, ja que el diari The Guardian és un diari britànic.

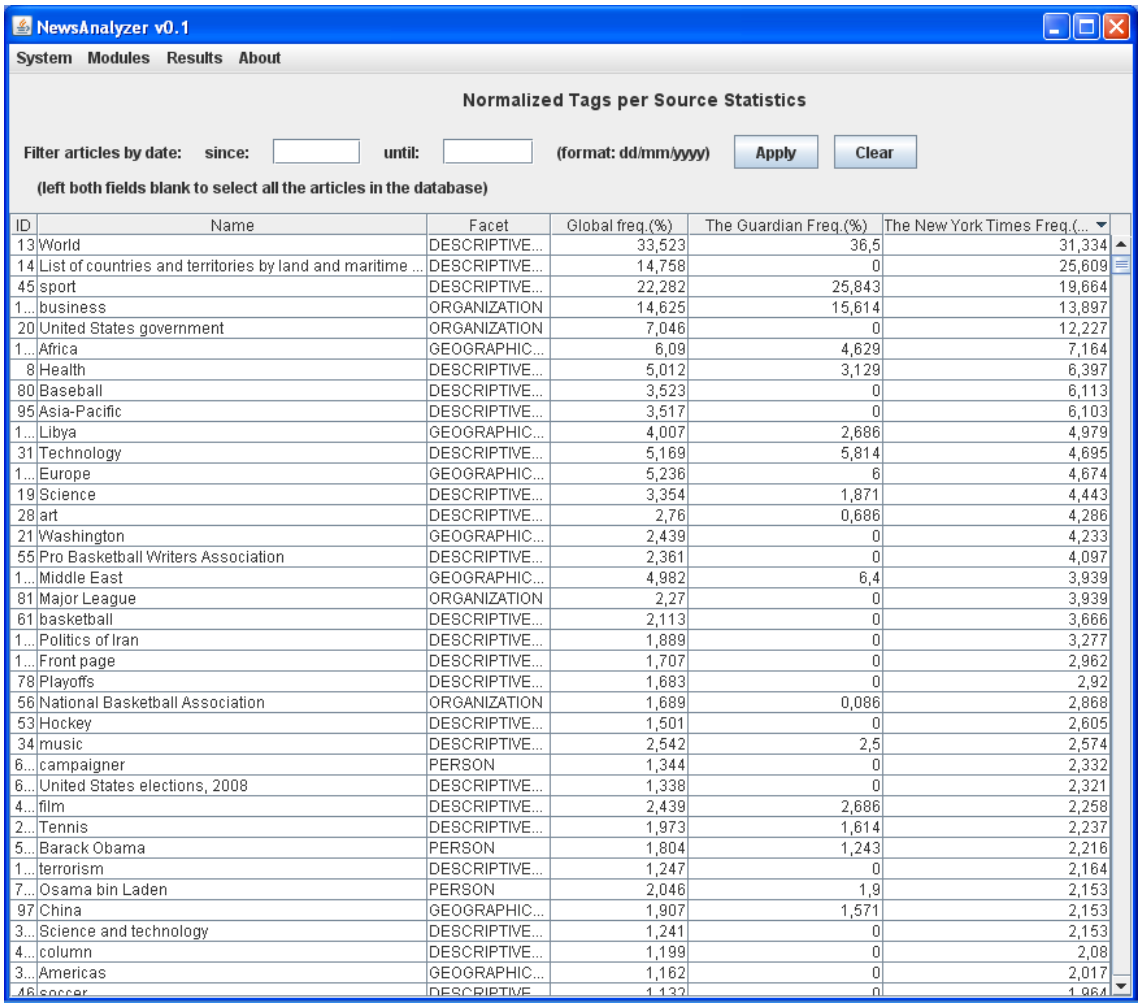
En segon lloc hi ha els esports, seguit dels negocis, política, mitjans de comunicació, societat, cultura...

És interessant veure que el futbol té més presència que la política. D'altra banda, no hi ha ni rastre del beisbol, com és normal, ja que no és un esport popular al regne unit.

Finalment, en l'àmbit internacional els Estats Units, l'Orient Mitjà i Europa superen en quantitat d'articles publicats a Àfrica.

7.3.3 Temàtiques més habituals al diari The New York Times

Si ara ens fixem en la freqüència de les etiquetes en el diari The New York Times observem el següent:



ID	Name	Facet	Global freq.(%)	The Guardian Freq.(%)	The New York Times Freq.(...
13	World	DESCRIPTIVE...	33,523	36,5	31,334
14	List of countries and territories by land and maritime ...	DESCRIPTIVE...	14,758	0	25,609
45	sport	DESCRIPTIVE...	22,282	25,843	19,664
1...	business	ORGANIZATION	14,625	15,614	13,897
20	United States government	ORGANIZATION	7,046	0	12,227
1...	Africa	GEOGRAPHIC...	6,09	4,629	7,164
8	Health	DESCRIPTIVE...	5,012	3,129	6,397
80	Baseball	DESCRIPTIVE...	3,523	0	6,113
95	Asia-Pacific	DESCRIPTIVE...	3,517	0	6,103
1...	Libya	GEOGRAPHIC...	4,007	2,686	4,979
31	Technology	DESCRIPTIVE...	5,169	5,814	4,695
1...	Europe	GEOGRAPHIC...	5,236	6	4,674
19	Science	DESCRIPTIVE...	3,354	1,871	4,443
28	art	DESCRIPTIVE...	2,76	0,686	4,286
21	Washington	GEOGRAPHIC...	2,439	0	4,233
55	Pro Basketball Writers Association	DESCRIPTIVE...	2,361	0	4,097
1...	Middle East	GEOGRAPHIC...	4,982	6,4	3,939
81	Major League	ORGANIZATION	2,27	0	3,939
61	basketball	DESCRIPTIVE...	2,113	0	3,666
1...	Politics of Iran	DESCRIPTIVE...	1,889	0	3,277
1...	Front page	DESCRIPTIVE...	1,707	0	2,962
78	Playoffs	DESCRIPTIVE...	1,683	0	2,92
56	National Basketball Association	ORGANIZATION	1,689	0,086	2,868
53	Hockey	DESCRIPTIVE...	1,501	0	2,605
34	music	DESCRIPTIVE...	2,542	2,5	2,574
6...	campaigner	PERSON	1,344	0	2,332
6...	United States elections, 2008	DESCRIPTIVE...	1,338	0	2,321
4...	film	DESCRIPTIVE...	2,439	2,686	2,258
2...	Tennis	DESCRIPTIVE...	1,973	1,614	2,237
5...	Barack Obama	PERSON	1,804	1,243	2,216
1...	terrorism	DESCRIPTIVE...	1,247	0	2,164
7...	Osama bin Laden	PERSON	2,046	1,9	2,153
97	China	GEOGRAPHIC...	1,907	1,571	2,153
3...	Science and technology	DESCRIPTIVE...	1,241	0	2,153
4...	column	DESCRIPTIVE...	1,199	0	2,08
3...	Americas	GEOGRAPHIC...	1,162	0	2,017
46	soccer	DESCRIPTIVE...	1,132	0	1,964

Figura 43 – Etiquetatge més abundant al diari The New York Times

Novament, les notícies internacionals ocupen la major part d'articles d'aquesta font, seguides de les esportives i de les de negocis.

En l'àmbit internacional Àfrica es troba en primer lloc, seguit d'Àsia, Líbia i Europa.

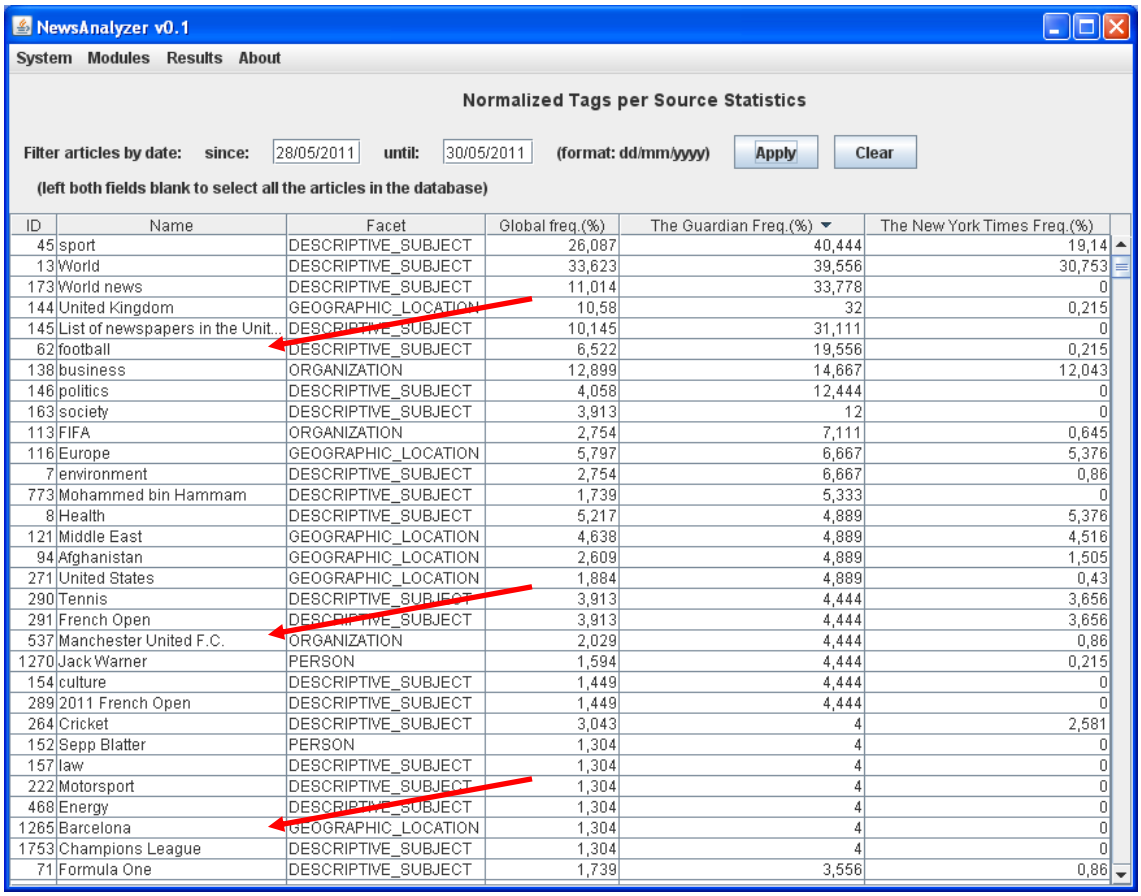
És interessant veure que l'esport més important sembla el beisbol seguit del basquet.

Finalment, sembla que a aquest diari li preocupen més els temes de salut que al britànic.

7.3.4 Temàtiques més en un període de temps concret

El programa newsAnalyzer incorpora la facilitat d'utilitzar un filtre per analitzar els temes de les publicacions en un període determinat de temps, fet que és molt interessant per determinar la importància que es dóna a determinada notícia durant aquest període.

Per exemple, entre les dates 28 i 29 de maig de 2011, en el diari The Guardian, el tema més important va ser l'esport:



ID	Name	Facet	Global freq.(%)	The Guardian Freq.(%)	The New York Times Freq.(%)
45	sport	DESCRIPTIVE_SUBJECT	26,087	40,444	19,14
13	World	DESCRIPTIVE_SUBJECT	33,623	39,556	30,753
173	World news	DESCRIPTIVE_SUBJECT	11,014	33,778	0
144	United Kingdom	GEOGRAPHIC_LOCATION	10,58	32	0,215
145	List of newspapers in the Unit...	DESCRIPTIVE_SUBJECT	10,145	31,111	0
62	football	DESCRIPTIVE_SUBJECT	6,522	19,556	0,215
138	business	ORGANIZATION	12,899	14,667	12,043
146	politics	DESCRIPTIVE_SUBJECT	4,058	12,444	0
163	society	DESCRIPTIVE_SUBJECT	3,913	12	0
113	FIFA	ORGANIZATION	2,754	7,111	0,645
116	Europe	GEOGRAPHIC_LOCATION	5,797	6,667	5,376
7	environment	DESCRIPTIVE_SUBJECT	2,754	6,667	0,86
773	Mohammed bin Hammam	DESCRIPTIVE_SUBJECT	1,739	5,333	0
8	Health	DESCRIPTIVE_SUBJECT	5,217	4,889	5,376
121	Middle East	GEOGRAPHIC_LOCATION	4,638	4,889	4,516
94	Afghanistan	GEOGRAPHIC_LOCATION	2,609	4,889	1,505
271	United States	GEOGRAPHIC_LOCATION	1,884	4,889	0,43
290	Tennis	DESCRIPTIVE_SUBJECT	3,913	4,444	3,656
291	French Open	DESCRIPTIVE_SUBJECT	3,913	4,444	3,656
537	Manchester United F.C.	ORGANIZATION	2,029	4,444	0,86
1270	Jack Warner	PERSON	1,594	4,444	0,215
154	culture	DESCRIPTIVE_SUBJECT	1,449	4,444	0
289	2011 French Open	DESCRIPTIVE_SUBJECT	1,449	4,444	0
264	Cricket	DESCRIPTIVE_SUBJECT	3,043	4	2,581
152	Sepp Blatter	PERSON	1,304	4	0
157	law	DESCRIPTIVE_SUBJECT	1,304	4	0
222	Motorsport	DESCRIPTIVE_SUBJECT	1,304	4	0
468	Energy	DESCRIPTIVE_SUBJECT	1,304	4	0
1265	Barcelona	GEOGRAPHIC_LOCATION	1,304	4	0
1753	Champions League	DESCRIPTIVE_SUBJECT	1,304	4	0
71	Formula One	DESCRIPTIVE_SUBJECT	1,739	3,556	0,86

Figura 44 – Resultats de l'anàlisi per a un període concret

Això és lògic, ja que va coincidir amb la final de la Champions Leage de la FIFA, que es jugava en el Regne Unit.

Els dos equips participants hi apareixen: el Manchester F.C. i un mal classificat Barcelona, en comptes de F.C. Barcelona. I és que aquest programa no és lliure d'errors, i cal millorar-lo molt encara.

7.3.5 Discussió dels resultats

Com s'ha pogut comprovar, es poden comparar fàcilment les temàtiques principals de cada font en els mateixos termes o similars i se'n poden extreure conclusions. A més a més, es poden fer anàlisis per a períodes concrets de temps, fet que permet visualitzar com es tracta una mateixa notícia per diferents fonts.

Finalment, convé recordar que els mètodes que emprats en aquest programa no són lliures d'error i que cal treballar molt encara per millorar-los i fer un tractament més intel·ligent de les etiquetes.

8 Conclusions i propostes de millora

Vistos i discutits els resultats obtinguts en aquest TFG, es descriuen tot seguit les conclusions a què s'ha arribat i les propostes de millora i treball futur que se'n deriven.

8.1 Conclusions

- La normalització de les etiquetes basada exclusivament en la informació que aporta el nom de la pròpia etiqueta és un procés complicat en el què inevitablement es produeixen resultats incorrectes. Tanmateix, la proporció de resultats favorables és prou elevada com per a poder utilitzar aquest mètode com una primera aproximació per als objectius del treball d'aquest TFG.

- La normalització ajudada mitjançant l'ús de la categoria a la qual pertany una etiqueta no pot basar-se exclusivament en la similitud de la jerarquia d'hypernyms entre l'etiqueta i la seva categoria, ja que en essència són dos conceptes diferents i no tenen perquè estar vinculats en la seva categorització, sinó en una relació semàntica molt més complexa.

- La categorització de les etiquetes en facets ha estat un procediment útil per comprovar que hi ha hagut millorar en el mètode de normalització, però no és una mesura quantitativa vàlida per a mesurar aquesta millora. Aquesta mesura no hi ha més remei que fer-la a mà o mitjançant conjunts d'etiquetes de prova.

- Respecte a la conclusió anterior, sí que pot ser molt interessant aprofitar la classificació en facets que ofereixen algunes fonts com el diari The New York Times per tal d'afinar més en el procés de normalització.

- Probablement, una bona estratègia per millorar la capacitat de normalització d'aquesta aplicació sigui la creació d'una ontologia pròpia a partir de les dades que es van obtenint amb una classificació jeràrquica de les etiquetes normalitzades que es vagi construint mitjançant algun procediment d'aprenentatge basat, per exemple, en les freqüències i probabilitats de coaparició de les etiquetes en els articles.

- Al contrari del que es pot arribar a pensar, els processos més limitants en el rendiment de l'aplicació són els d'accés a la base de dades, i no els d'accés a la xarxa Internet, donat a la gran quantitat de dades que es generen en les consultes, en el cas dels recol·lectors de notícies, i a la gran quantitat d'interrelacions entre etiquetes, etiquetes normalitzades i notícies, en el cas de l'anàlisi.

- DBpedia és una eina molt potent per a la classificació de conceptes quan WordNet falla, tot i que no està lliure d'ambigüitats i no ofereix una jerarquia nítida d'hypernyms.

- WordNet és una eina molt potent d'anàlisi semàntic de la qual se'n pot treure molt més profit en un futur per a aquesta aplicació o similars.

8.2 Propostes de millora

- Desenvolupar un procediment de normalització que tingui en compte la categoria a la què pertany l'etiqueta per tal de millorar l'encert en la tria de l'accepció corresponent entre els diferents candidats en el procediment de normalització. Aquest procediment podria basar-se en l'anàlisi de la definició de l'etiqueta i de la categoria, ja que la jerarquia d'hypernyms s'ha demostrat que no és la millor estratègia.

- Desenvolupar un procediment d'aprenentatge que permeti categoritzar i classificar les notícies en categories més que no pas en facets, de tal manera que les categories obtingudes podrien ajudar en el procediment de normalització. Aquest procediment d'aprenentatge es pot basar en la freqüència i probabilitat de coaparició de les etiquetes en els articles.

- Utilitzar mètriques de similitud entre frases per tal de millorar el procediment de selecció del millor candidat en la normalització quan no hi hagi categoria.

- Un cop vist que el procediment de normalització millora, ja no té sentit classificar les etiquetes en facets amb el propòsit d'avaluar el procediment de normalització. En tot cas només caldria mantenir-los si es vol emprar aquesta classificació per millorar la capacitat de possibles futures cerques. Més que aquest tipus de classificació, el més interessant seria utilitzar els facets provinents de les fonts de notícies per millorar el procediment de normalització.

- Millorar els procediments de recollida per tal de fer-la més flexible.

- Millorar el procediment d'anàlisi i permetre diferents anàlisis i que es puguin visualitzar per separat, ja que DBpedia i Wikipedia són fonts dinàmiques.

- Revisar les estructures de dades per incrementar l'eficiència del programa. Ja s'estan utilitzant moltes taules de hash i conjunts ordenats, però no és suficient.

9 Bibliografia

La relació de fonts emprades en aquest TFG ha estat la següent:

1. **Bataller Díaz, A., et al.** (2008) *Treball final de carrera* (1a edició). Barcelona: Fundació UOC. ISBN: 978-84-691-4809-9.

2. **Borort, S.** (2009) *Semantic classification of social tags for faceted search* [Treball final de màster]. Thailand: Asian Institute of Technology. School of Engineering and Technology.
   [Data de consulta: 7 de juny de 2011]
   <http://www.cs.ait.ac.th/~b106252/files/thesis/>

3. **Campderrich Falgueras, M.** (2004). *Enginyeria del programari* (1a edició). Barcelona: Fundació UOC. ISBN: 84-9788-065-X.

4. **Guardian, The** (2010, 20 de maig)
   "Content API: Content Search Reference Guide".
   Open Platform. Build applications with the guardian.
   [Document de referència en línia]
   [Data de consulta: 30 maig de 2011]
   <http://www.guardian.co.uk/open-platform/content-api-content-search-reference-guide>

5. **json-simple** (2009, 23 de gener)
   "project home"
   JSON.simple – A simple Java toolkit for JSON
   [Pàgina web del projecte json-simple]
   [Data de consulta: 7 de juny de 2011]
   <http://code.google.com/p/json-simple/>

6. **JSON.org** (?)
   Introducing JSON
   [Documentació sobre el format JSON]
   [Data de consulta: 7 de juny de 2011]
   <http://www.json.org/>

7. **MediaWiki** (2011, 5 de juny)
   "API:Main page"
   MediaWiki.org
   [Documentació de l'API]
   [Data de consulta: 7 de juny de 2011]
   <http://www.mediawiki.org/wiki/API>

8. **New York Times, The** (2010, 31 d'agost)
   "API documentation and tools"
   The New York Times Developer Network Beta
   [Document de referència en línia]

[Data de consulta: 30 de maig de 2011]
<http://developer.nytimes.com/docs>

9. **Pew Research Center for the People & the Press** (2011, 14 de març)
"Overview: key findings".
The State of the News Media 2011
[Report en línia].
[Data de consulta: 30 maig de 2011]
<http://stateofthemedia.org/2011/overview-2/key-findings/>

10. **Pradel Miquel, J., Raya Martos, J.** (2001) *Enginyeria del programari* (1a edició).
Barcelona: Fundació UOC. ISBN: 978-84-693-9175-4.

11. **Princeton University** (2010, 27 d'octubre)
"WordNet Documentation"
WordNet – A lexical database for English
[Documentació online]
[Data de consulta: 7 de juny de 2011]
<http://wordnet.princeton.edu/wordnet/documentation/>

12. **Prud'hommeaux, E., Seaborne, A.** (2008, 15 de gener).
SPARQL Query Language for RDF
[Article/Documentació]
[Data de consulta: 7 de juny de 2011]
<http://www.w3.org/TR/rdf-sparql-query/>

13. **Spell, B**. (2009, 24 de desembre)
Java API for WordNet Searching (JAWS)
[Pàgina web del projecte JAWS amb la seva documentació]
<http://lyle.smu.edu/~tspell/jaws/index.html>

14. **Sun Microsystems, Inc.** (2001-2002)
"Core J2EE Patterns - Data Access Object"
Core J2EE Pattern Catalog
[Llibre electrònic]
Oracle Corporation and/or its affiliates
[Data de consulta: 7 de juny de 2011]
<http://java.sun.com/blueprints/corej2eepatterns/Patterns/DataAccessObject.html>

15. **Thibodeau, T. Jr** (2011, 4 de maig)
DBpedia
[Pàgina web oficial de DBpedia amb tot tipus de documentació]
[Data de consulta: 7 de juny de 2011]
<http://dbpedia.org/About>

Annexos

Annex A – Mitjans emprats

- Sistema operatiu de desenvolupament: **Microsoft Windows XP Service Pack 3**
- Entorn Integrat de Desenvolupament: **NetBeans IDE**
- Entorn de desenvolupament de la base de dades: **MySQL Workbench**
- Edició de textos: **Microsoft Office 2003**
- Edició de diagrames: **Dia**

Annex B – Manual de compilació del programa

1) Assegurar-se de tenir instal·lat el Java SE Development Kit 6 (JDK 6)

   L'enllaç de la web oficial de descàrrega és el següent:

   http://www.oracle.com/technetwork/java/javase/downloads/index.html

2) Assegurar-se de tenir incorporat el directori *bin* del JDK al PATH del sistema

   A Microsoft Windows XP:

   - Accedir al *Tauler de control* (Menú *Inicia* →*Configuració* → *Tauler de control*) i un cop allí accedir a la funcionalitat *Sistema*.
   - En la pantalla de Propietats del sistema accedir a la pestanya *Avançats* i prémer el botó *Variables d'entorn*.
   - En l'apartat de *Variables del sistema* editar l'entrada *Path* i afegir al final del valor actual la ruta cap al directori *bin* del JDK precedida d'un ; (punt i coma).

3) Disposar del connector JDBC 5.x de MySQL

   L'enllaç de la web oficial de descàrrega és el següent:

   http://dev.mysql.com/downloads/connector/j/

   Aquest fitxer també es proporciona amb l'aplicació al directori *externalLibs*.

4) Disposar de la llibreria json-simple 1.1

   L'enllaç de la web oficial de descàrrega és el següent:

   http://code.google.com/p/json-simple/downloads/list

   Es recomana descarregar el fitxer *json_simple-1.1.jar*

   Aquest fitxer també es proporciona amb l'aplicació al directori *externalLibs*.

5) Disposar de la llibreria Java API for WordNet Searching (JAWS) versió 1.3

   L'enllaç de la web oficial de descàrrega és el següent:

   http://lyle.smu.edu/~tspell/jaws/index.html#downloads

   Es recomana descarregar el fitxer *jaws-bin.jar*

Aquest fitxer també es proporciona amb l'aplicació al directori *externalLibs*.

6) Tenir incorporats el JDBC de MySQL i les llibreries JSON simple i JAWS en el *Classpath* del sistema.

   A Microsoft Windows:

   - Accedir al *Tauler de control* (Menú *Inicia* →*Configuració* → *Tauler de control*) i un cop allí accedir a la funcionalitat *Sistema*.
   - En la pantalla de Propietats del sistema accedir a la pestanya *Avançats* i prémer el botó *Variables d'entorn*.
   - En l'apartat de *Variables del sistema* editar l'entrada *CLASSPATH* i afegir al final del valor actual les rutes cap als diferents fitxers precedides d'un ; (punt i coma).

7) Executar el fitxer *compile.bat*

   La compilació de l'aplicació corre a càrrec del script *compile.bat*. Aquest script es troba a l'arrel del projecte i s'encarrega de:

   1) Crear el directori *dist*, on hi haurà tot el contingut del lliurable de l'aplicació.
   2) Fer les crides corresponents al compilador per tal de compilar totes les classes i desar els fitxers *.class* als directoris corresponents dins del *dist*.
   3) Copiar tots els fitxers *.properties*, que contenen les diferents opcions de configuració del programa, als directories corresponents del *dist*.
   4) Copiar el directori *sql*, que conté els scripts per a la creació de la base de dades i de l'usuari corresponent, al directori *sql* dins del *dist*.
   5) Generar el script *run.bat*, que permetrà executar l'aplicació, dins el directori *dist*.
   6) Generar el script *run.bat* en el directori arrel del projecte, que fa una crida al *run.bat* de l'apartat 5 per tal d'executar el programa des de l'arrel de projecte.
   7) Generar el fitxer compile_log.txt amb tots els warnings i errors que s'hagin pogut produir en el procediment de compilació.

   En resum, en executar aquest fitxer es generarà el següent contingut:

   - Directori *dist*: conté tots els fitxers que composen el lliurable de l'aplicació. Conté les classes compilades, els fitxers .properties de configuració, les llibreries externes necessàries per executar l'aplicació i un script anomenat *run.bat* que permet executar l'aplicació.
   - Fitxer *compile_log.txt*: conté els missatges de warning i d'error que generi el compilador.
   - Fitxer *run.bat*: petit script que executa el fitxer run.bat que es troba en la carpeta *dist*.

Annex C – Manual d'instal·lació de l'aplicació

1) Excepte l'existència del JDK, la resta de requisits per compilar el programa ho són també per executar-lo, així que cal assegurar-se que es compleixen els punts de la compilació del 3 al 6 (ambdós inclosos).

2) Cal disposar del Java Runtime Environment per tal d'executar el programa. Aquest es més lleuger que el JDK i es pot descarregar de la mateixa pàgina. Tanmateix, si ja es té instal·lat el JDK, aquest també incorpora el JRE, amb la qual cosa no serà necessari instal·lar-lo.

3) Instal·lar el sistema gestor de bases de dades MySQL Community Server 5.x

   L'enllaç de la web oficial de descàrrega és el següent:

   http://dev.mysql.com/downloads/mysql/

   A Microsoft Windows XP es pot instal·lar el programa com a servei i fer que s'iniciï automàticament amb el sistema.

   Una bona eina gràfica per a la gestió de bases de dades MySQL és el MySQL Workbench, que es pot descarregar de la mateixa pàgina que el Community Server.

   Alternativament a la instal·lació del server i del workbench, es pot descarregar la darrera versió portable de XAMPP, que permet gestionar de forma fàcil bases de dades MySQL i administrar-les mitjançant l'entorn web phpMyAdmin.

4) Carregar els fitxers *NewsAnalyzerDDL.sql* i *NewsAnalyzerUserDDL* que es troba en el directori *sql* de l'aplicació

   - En el cas d'emprar MySQL Workbench, en l'apartat *SQL Development* crear una nova connexió (*New Connection*) a la base de dades mitjançant l'usuari *root* o un usuari de privilegis similars.
   - Connectar-se a la base de dades mitjançant la connexió creada, carregar el fitxer *NewsAnalyzerDDL.sql* (*File → Open SQL Script...*) i executar-lo (*Query → Execute (All or Selection)*)
   - Fer el mateix amb el fitxer *NewsAnalyzerUserDDL.sql*

5) Configurar l'accés a la base de dades

   Dins el directori newsAnalyzer que conté els fitxers compilats, editar el fitxer NewsAnalyzerProperties.properties i incorporar els diferents paràmetres de connexió a la base de dades. Per exemple:

```
#The Database Connection Properties
```

```
DBEngine=MySQL
DBHost=127.0.0.1
DBPort=3306
DBName=newsAnalyzer
DBUser=newsAnalyzer
DBPassword=newsAnalyzer
```

6) Instal·lar WordNet

L'enllaç de la web oficial de descàrrega és el següent:

http://wordnet.princeton.edu/wordnet/download/

Per a Microsoft Windows XP es recomana la versió 2.1

7) Configurar el JAWS

Dins el directori newsAnalyzer que conté els fitxers compilats, editar el fitxer NewsAnalyzerProperties.properties i incorporar la ruta cap al directori dict de WordNet en l'entrada *wordnet.database.dir*. El separador entre directoris és una doble contrabarra. A tall d'exemple:

```
wordnet.database.dir=C:\\Archivos de programa\\WordNet\\2.1\\dict
```

8) Afegir el "directori actual" al CLASSPATH

Accedir la variable de sistema CLASSPATH de la mateixa manera que es va realitzar en el punt 6 de la compilació. Al final del valor actual d'aquesta variable afegir el següent:

;.

(*punt i coma* seguit d'un *punt*)

Annex D – Manual d'usuari

1. Execució

Per tal d'executar el programa únicament cal executar el fitxer *run.bat*. Tot seguit, mitjançant el manual d'usuari, es detallen les diferents funcionalitats que ofereix el programa a través de la seva GUI.

En iniciar l'aplicació mitjançant el script run.bat es mostrarà la següent pantalla:

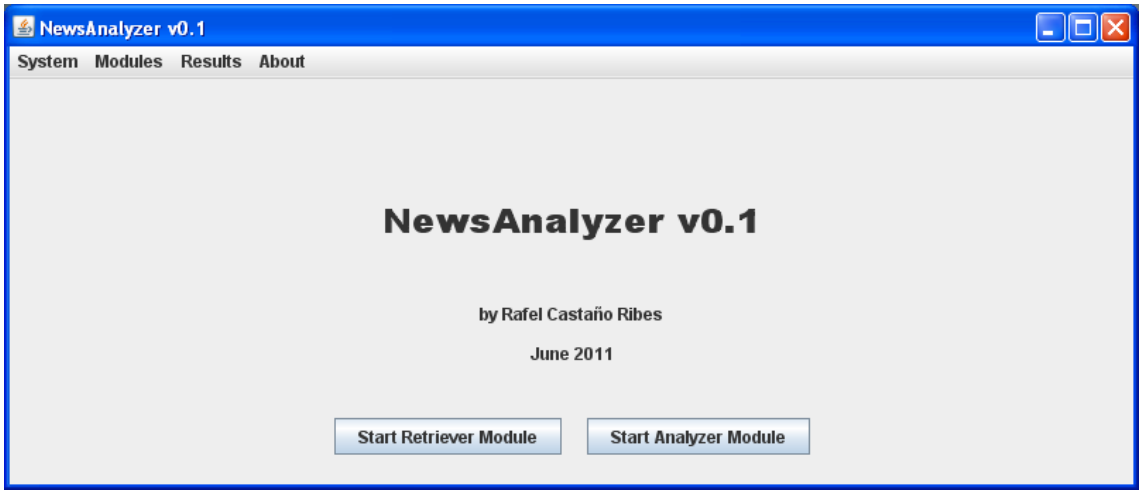


Figura 45 – Vista de la pantalla principal

Aquesta pantalla permet iniciar totes les funcionalitats del programa a través dels seus menús. Tot seguit es descriuen les accions que permeten dur a terme cadascun dels menús.

2. Relació de menús i les seves funcionalitats

- System
    - Exit

      Permet sortir del programa, no finalitza les tasques programades als mòduls recol·lector i analitzador. Per finalitzar aquestes tasques cal tancar les seves respectives pantalles.

- Modules
    - Start Retriever Module

      Obre la pantalla de configuració del mòdul recol·lector de notícies. En ella es pot configurar des de quina data es volen recol·lectar notícies i si es vol que el programa les vaig recol·lectant automàticament a partir d'ara i amb quina freqüència. Aquesta pantalla utilitza un patró singleton per a instanciar-se,

amb la qual cosa només en podrà haver una per a cada màquina virtual de java que executi el programa.

Si la pantalla es tanca mitjançant el botó X, les recol·leccions iniciades s'aturen.

Figura 46 – Vista de la pantalla del recol·lector de notícies

- *Retrieve articles since*: indica la data des de la qual es vol iniciar la recol·lecta de notícies. Cal que estigui en format dd/mm/yyyy. De tota manera, fins que no s'introdueix una data vàlida no es permetrà que s'activi el botó *Retrieve* per iniciar la recol·lecta. Convé que la data d'inici de la recol·lecció no sigui molt anterior a l'actual ja que la majoria d'APIs de recol·lecció de notícies tenen una limitació en la quantitat d'articles que es poden recollir al dia i aquesta excepció encara no està ben controlada per l'aplicació (per manca de temps).

- *Select article sources*: permet seleccionar una o més fonts per a la recol·lecta de notícies. Aquestes fonts les detecta el sistema automàticament si s'han editat correctament els fitxers de configuració tal i com es descriu en el manual d'instal·lació dels mòduls recol·lectors. Fins que no se selecciona al menys una font no es permetrà que s'activi el botó *Retrieve* per iniciar la recol·lecta.

- *Automatically retrieve*: cal marcar aquesta opció si es vol que el recol·lector de notícies no s'aturi en arribar al moment actual i que continuï recol·lectant al llarg del temps, altrament, en arribar a la data actual s'aturarà. Si se selecciona aquesta opció es mostrarà una caixa de text on cal introduir cada quants minuts es desitja que es dugui a terme la recol·lecció. Fins que no s'omple correctament aquesta caixa no es permetrà que s'activi el botó *Retrieve* per iniciar la recol·lecta.

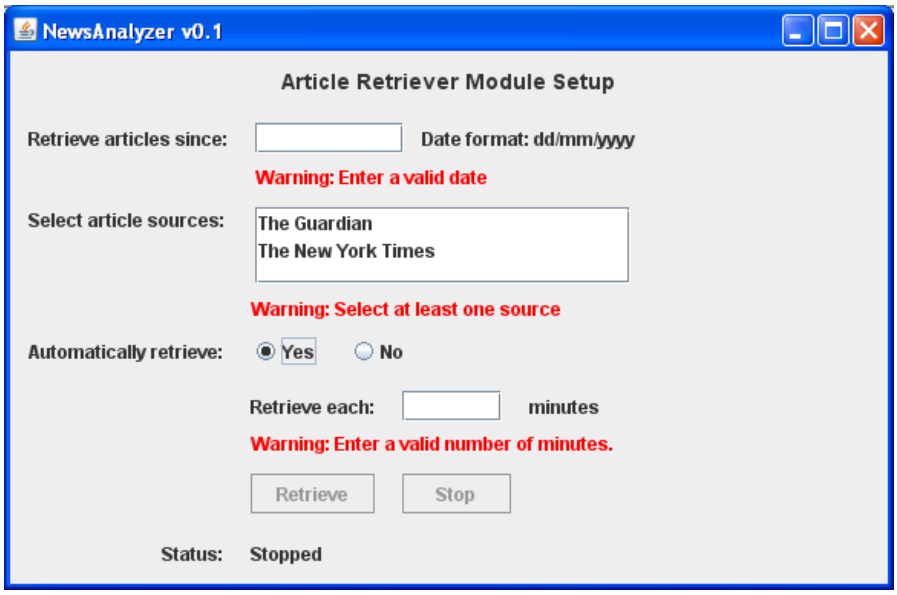


Figura 47 – Vista de les opcions de configuració de la recol·lecció automàtica

- *Retrieve*: botó que permet iniciar la recol·lecta. Aquest botó només s'activa quan tots els camps s'omplen de forma satisfactòria.

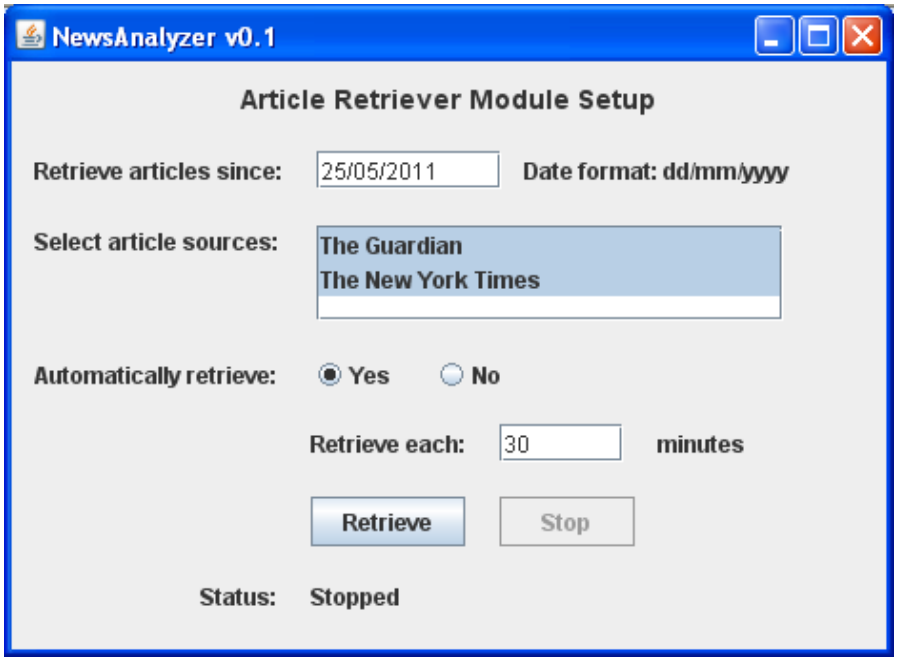


Figura 48 – Exemple de configuració per al recol·lector

- *Stop*: botó que permet aturar la recol·lecta automàtica. Aquest botó només s'activa quan s'inicia una recol·lecció automàtica.

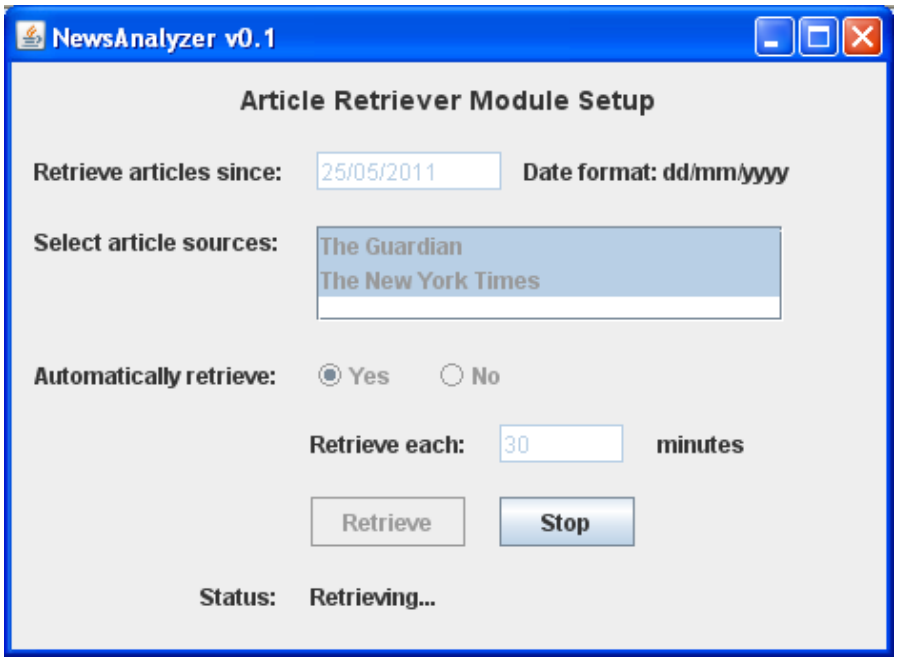


Figura 49 – Vista de la pantalla del recol·lector durant la recol·lecció

- Start Analyzer Module

  Obre la pantalla de configuració del mòdul analitzador d'etiquetes. En ella es pot configurar si es volen iniciar els anàlisis automàtics i cada quan es vol que s'analitzi la informació pendent d'analitzar.

  Si la pantalla es tanca mitjançant el botó X, els anàlisis iniciats s'aturen.

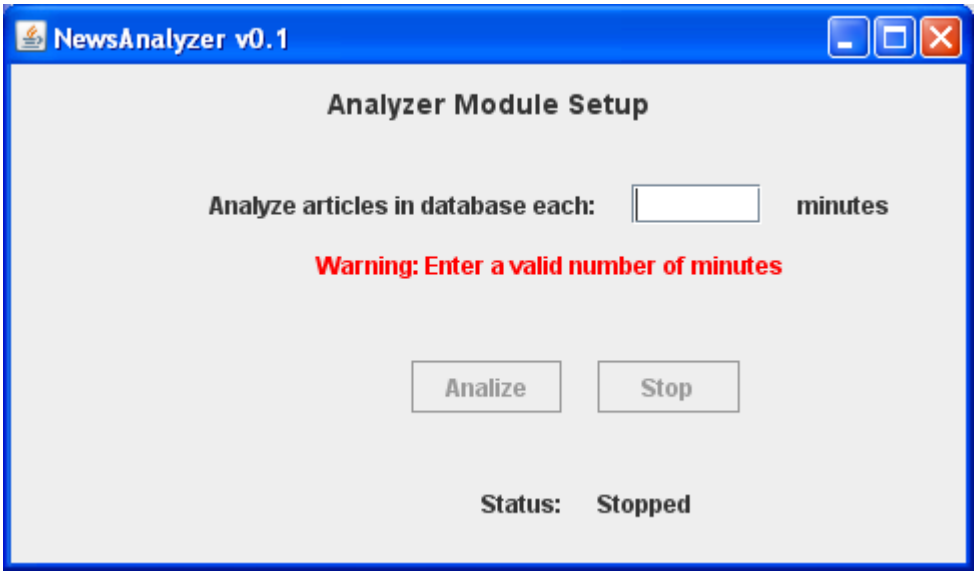


Figura 50 – Vista de la pantalla de l'analitzador

  - *Analyze articles in database each*: permet definir cada quants minuts es vol que el programa faci una anàlisi de les etiquetes pendents d'analitzar. El botó analyze no s'activa fins que no hi ha un valor vàlid en aquest camp.

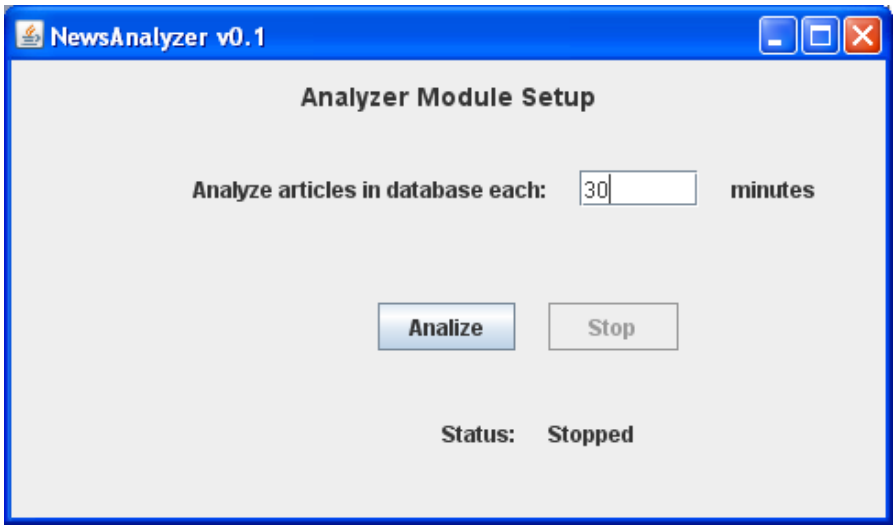


Figura 51 – Exemple de configuració de l'analitzador

- *Analyze*: inicia l'anàlisi automàtic.

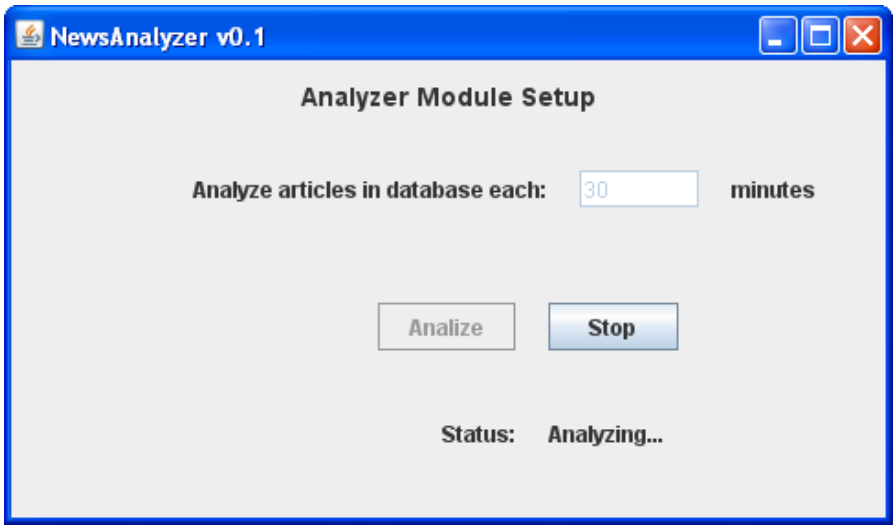


Figura 52 – Vista de la pantalla de l'analitzador durant l'anàlisi

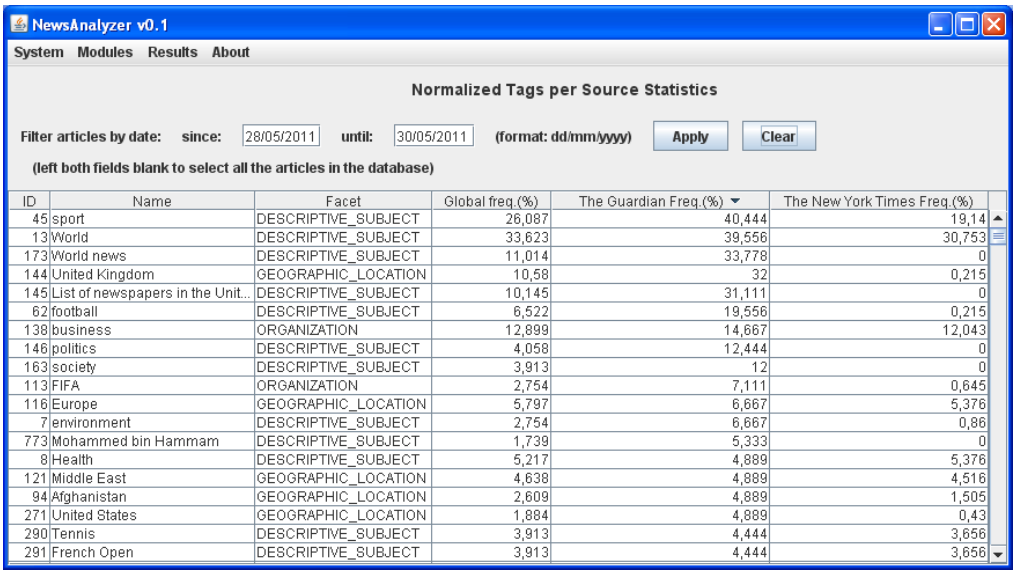
- *Stop*: atura l'anàlisi automàtic.

- Results

  - Show Normalized Tags per Source Statistics

    Mostra una taula resum amb els percentatges d'aparició de cadascun dels tags normalitzats per a cada font de notícies. Això permet fer una anàlisi de quines són les temàtiques més habituals per a cada font.

    Es permet ordenar les files en funció dels camps fent clic a sobre del nom del propi camp. D'aquesta manera es pot visualitzar quins són els tags normalitzats més freqüents per a cada font.



ID	Name	Facet	Global freq.(%)	The Guardian Freq.(%)	The New York Times Freq.(%)
45	sport	DESCRIPTIVE_SUBJECT	26,087	40,444	19,14
13	World	DESCRIPTIVE_SUBJECT	33,623	39,556	30,753
173	World news	DESCRIPTIVE_SUBJECT	11,014	33,778	0
144	United Kingdom	GEOGRAPHIC_LOCATION	10,58	32	0,215
145	List of newspapers in the Unit...	DESCRIPTIVE_SUBJECT	10,145	31,111	0
62	football	DESCRIPTIVE_SUBJECT	6,522	19,556	0,215
138	business	ORGANIZATION	12,899	14,667	12,043
146	politics	DESCRIPTIVE_SUBJECT	4,058	12,444	0
163	society	DESCRIPTIVE_SUBJECT	3,913	12	0
113	FIFA	ORGANIZATION	2,754	7,111	0,645
116	Europe	GEOGRAPHIC_LOCATION	5,797	6,667	5,376
7	environment	DESCRIPTIVE_SUBJECT	2,754	6,667	0,86
773	Mohammed bin Hammam	DESCRIPTIVE_SUBJECT	1,739	5,333	0
8	Health	DESCRIPTIVE_SUBJECT	5,217	4,889	5,376
121	Middle East	GEOGRAPHIC_LOCATION	4,638	4,889	4,516
94	Afghanistan	GEOGRAPHIC_LOCATION	2,609	4,889	1,505
271	United States	GEOGRAPHIC_LOCATION	1,884	4,889	0,43
290	Tennis	DESCRIPTIVE_SUBJECT	3,913	4,444	3,656
291	French Open	DESCRIPTIVE_SUBJECT	3,913	4,444	3,656

Figura 53 – Vista de la pantalla de resultats per font

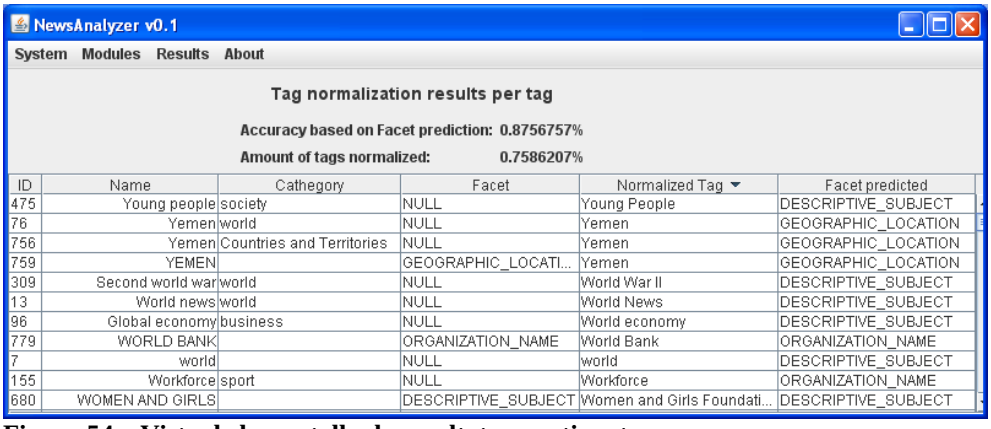
També es permet filtrar els resultats per a un període determinat de temps. En absència de filtre es mostren els resultats per a tots els articles existents a la base de dades.

Aquesta pantalla es pot millorar posant un tabbed pane amb cada font en un tab diferent. D'aquesta manera quedarien totes les etiquetes ordenades de major a menor aparició per a cada font.

- Show Tag Normalization Results per Tag

  Mostra el resultat del procediment d'anàlisi (normalització + classificació dels tags), així com el percentatge d'encert en la predicció dels facets i el percentatge d'etiquetes que s'han pogut normalitzar fins al moment.

  Aquesta pantalla triga una estona en carregar-se.



ID	Name	Cathegory	Facet	Normalized Tag	Facet predicted
475	Young people	society	NULL	Young People	DESCRIPTIVE_SUBJECT
76	Yemen	world	NULL	Yemen	GEOGRAPHIC_LOCATION
756	Yemen	Countries and Territories	NULL	Yemen	GEOGRAPHIC_LOCATION
759	YEMEN		GEOGRAPHIC_LOCATI...	Yemen	GEOGRAPHIC_LOCATION
309	Second world war	world	NULL	World War II	DESCRIPTIVE_SUBJECT
13	World news	world	NULL	World News	DESCRIPTIVE_SUBJECT
96	Global economy	business	NULL	World economy	DESCRIPTIVE_SUBJECT
779	WORLD BANK		ORGANIZATION_NAME	World Bank	ORGANIZATION_NAME
7	world		NULL	world	DESCRIPTIVE_SUBJECT
155	Workforce	sport	NULL	Workforce	ORGANIZATION_NAME
680	WOMEN AND GIRLS		DESCRIPTIVE_SUBJECT	Women and Girls Foundati...	DESCRIPTIVE_SUBJECT

Figura 54 – Vista de la pantalla de resultats per etiqueta

- About
  - About info

Mostra informació sobre la versió del programa, l'autor, la data de publicació i un parell de botons per si es volen iniciar les pantalles corresponents de configuració i inici dels mòduls de recol·lecció i anàlisi.

És la mateixa pantalla que es mostra en iniciar l'aplicació.

Figura 55 – Vista de la pantalla d'informació general

3. Mode debug

El mode debug permet que es mostri missatges de text per la sortida estàndard (p.e. la consola MS DOS en entorns Microsoft Windows) tot indicant en tot moment quina acció s'està duent a terme.

Per indicar a l'aplicació que es vol executar en mode debug cal modificar el valor de la propietat debug en el fitxer NewsAnalyzerProperties.properties i posar-hi valor true:

```
#Debug mode
debug=true
```

Per inhabilitar el mode debug només cal posar la propietat a false.

Annex E – Índex de figures

Figura 1 – Estructuració del treball 4
Figura 2 – Taula resum de les tasques del TFG 7
Figura 3 – Diagrama de Gantt amb la planificació del TFG 7
Figura 4 – Sol·licitud HTTP cap a l'API del diari The Guardian..... 9
Figura 5 – Part de la resposta de l'API del The Guardian en format JSON 10
Figura 6 – Proposta de normalització d'etiquetes (Borort, S.; 2009). 11
Figura 7 – Proposta de classificació d'etiquetes (Borort, S.; 2009). 12
Figura 8 – Diagrama de casos d'ús de NewsAnalyzer..... 16
Figura 9 – Diagrama de classes del domini de NewsAnalyzer 19
Figura 10 – Diagrama de col·laboració del cas d'ús "Iniciar el recol·lector"..... 21
Figura 11 – Diagrama de col·laboració del cas d'ús "Iniciar l'analitzador" 21
Figura 12 – Diagrama de col·laboració del cas d'ús "Visualitzar resultats per font" 22
Figura 13 – Diagrama de col·laboració del cas d'ús "Visualitzar resultats per etiqueta" 22
Figura 14 – Model de pantalla principal amb el menú de l'aplicació..... 23
Figura 15 – Opcions del menú System..... 23
Figura 16 – Opcions del menú Modules..... 23
Figura 17 – Opcions del menú Results..... 23
Figura 18 – Opcions del menú About..... 24
Figura 19 – Pantalla ArticleRetrieverGUI..... 24
Figura 20 – Pantalla AnalyzerGUI..... 24
Figura 21 – Pantalla ResultsPerSourceGUI..... 25
Figura 22 – Pantalla ResultsPerTagGUI 25
Figura 23 – Diagrama Entitat-Relació de NewsAnalyzer 29
Figura 24 – Implementació del patró DAO: diagrama de classes resultant 31
Figura 25 – Diagrama UML del mòdul recol·lector (retrievers) 33
Figura 26 – Diagrama UML del mòdul analitzador (analyzer)..... 39
Figura 27 – Diagrama d'activitat del mètode principal d'AnalyzerModule 40
Figura 28 – Diagrama d'activitat del mètode de normalització 41
Figura 29 – Jerarquia d'hypernyms del mot *United Kingdom* 42
Figura 30 – Taula resum d'assignació de facets des de WordNet 42
Figura 31 – Valors del camp rdf :type per a Paul McCartney a la DBpedia..... 42
Figura 32 – Resultats amb l'algorisme de normalització de referència 43
Figura 33 – Resultats amb l'algorisme de normalització modificat*..... 43
Figura 34 – Resposta de l'API de Wikipedia al mot 'Obama'..... 44
Figura 35 – Diagrama UML del mòdul explorador de resultats (resultsExplorer) 51
Figura 36 – Quantitat promig d'articles per font i dia..... 55
Figura 37 – Quantitat promig d'assignacions d'etiquetes per font i dia..... 55
Figura 38 – Resultats obtinguts en la normalització amb el mètode de referència 57
Figura 39 – Resultats de la classificació en facets amb el mètode de referència 58
Figura 40 – Resultats de la normalització amb el mètode millorat 59
Figura 41 – Etiquetatge normalitzat global més abundant 61
Figura 42 – Etiquetatge més abundant al diari The Guardian 62
Figura 43 – Etiquetatge més abundant al diari The New York Times 63
Figura 44 – Resultats de l'anàlisi per a un període concret..... 64
Figura 45 – Vista de la pantalla principal..... 74
Figura 46 – Vista de la pantalla del recol·lector de notícies..... 75
Figura 47 – Vista de les opcions de configuració de la recol·lecció automàtica..... 76

Figura 48 – Exemple de configuració per al recol·lector 76
Figura 49 – Vista de la pantalla del recol·lector durant la recol·lecció 77
Figura 50 – Vista de la pantalla de l'analitzador 77
Figura 51 – Exemple de configuració de l'analitzador 78
Figura 52 – Vista de la pantalla de l'analitzador durant l'anàlisi 78
Figura 53 – Vista de la pantalla de resultats per font 79
Figura 54 – Vista de la pantalla de resultats per etiqueta 79
Figura 55 – Vista de la pantalla d'informació general 80