Títol: **BibTEX Bibliography Index Maker**

Volum: **1/1**
Alumne: **Ramon Xuriguera Albareda**

Director/Ponent: **Marta Arias**
Departament: **LSI**
Data: **Primavera 2010**

DADES DEL PROJECTE

Títol del Projecte:

Nom de l'estudiant: Ramon Xuriguera Albareda

Titulació: Enginyeria Informàtica

Crèdits: 37,5

Director/Ponent: Marta Arias

Departament: LSI

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President:

Vocal:

Secretari:

QUALIFICACIÓ

Qualificació numèrica:

Qualificació descriptiva:

Data:

Índex

Capítol 1

Introducció

El format PDF ha esdevingut un estàndar per la divulgació de publicacions on-line.

Actualment existeixen serveis com ara *Google Scholar*, *Microsoft Academic Search* o *CiteSeer* que es dediquen a recol·lectar informació sobre articles i que comptabilitzen les referències entre diferents publicacions. En el cas de *CiteSeer*, al ser un projecte lliure, sabem que funciona analitzant les diferents parts dels articles, com ara les cites, però que també té problemes per obtenir els camps de la capçalera, que és el que ens interessa [GBL98].

Per una altra banda, també existeixen nombroses aplicacions dedicades al maneig de referències com *JabRef*[1] o *Mendeley*[2]. Entre algunes de les funcionalitats addicionals que ofereixen, hi ha la possibilitat de cercar en bases de dades d'articles i utilitzar les meta-dades dels fitxers per tal de trobar informació com ara el títol o l'autor. A banda d'això, no n'hem trobat cap que aprofiti el contingut dels documents per generar la referència.
El nostre sistema mira d'omplir el buit que queda entre aquestes dos tipus de programari que *BibTEX Bibliography Index Maker* és una eina d'ajuda a la creació d'índexs bibliogràfics pensada com un complement a aplicacions de maneig de referències ja existents com poden ser .

La principal funcionalitat que ofereix consisteix en escanejar un directori que conté articles científics en PDF i generar un índex bibliogràfic en BibTEX amb les referències d'aquests fitxers. Aquest índex es pot importar des de les aplicacions esmentades o bé pot ser referenciat directament des d'un nou document TEX.
Compta amb tres parts principals:

1.1 Estructura d'aquest document

La memòria està dividida en els següents capítols:

- Capitol 2: Descripció formal del projecte i el seu context així com un repàs sobre el disseny.
- Capitol 3: Parla de les tècniques que s'utilitzen per poder aconseguir pàgines que continguin informació sobre els do

[1] http://jabref.sourceforge.net
[2] http://www.mendeley.com

- Capitol 4: Tracta sobre com extreure la informació sobre els articles de les pàgines d'Internet que hem obtingut.
- Capitol 5: Està dedicat a les tècniques per generar, de forma automàtica, les regles d'extracció de les referències.
- Capitol 6: Plantejament de les proves per cadascuna de les parts més importants del sistema i anàlisi dels resultats obtinguts.

Pel que fa al contingut de les diferents seccions, ens hem volgut centrar, sobretot, en el que fa a les decisions que hem hagut de prendre al llarg de la realització del projecte i no tant en els aspectes tècnics de com s'ha implementat el sistema. Per cada decisió es presenten algunes de les opcions plantejades en un principi, els problemes que aquestes presenten i es passa a descriure la solució escollida i els motius de l'elecció.

Capítol 2

Definició del Projecte

2.1 Context

2.2 BibTEX

Per poder entendre el context del projecte cal que descrivim l'eina de maneig de referències BibTEX i la sintaxi del llenguatge que utilitza. En el nostre cas farem servir aquest llenguatge com a format de sortida al generar els índexos bibliogràfics. Al llistat 2.1 es mostra un exemple d'una referència d'un article científic expressat en el format BibTEX:

```
@article{MoSh:27,
  title = {Size direction games over the real line},
  author = {Moran, Gadi and Shelah, M., Saharon},
  journal = {Israel Journal of Mathematics},
  pages = {442--449},
  volume = {14},
  year = {1973},
}
```

Llistat 2.1: Referència expressada en BibTEX

Alguns aspectes a comentar sobre l'exemple anterior:

- La primera línia conté el tipus de document i un identificador. El primer defineix els camps obligatoris que s'han d'especificar, i el segon ens permetrà citar a la referència des d'un document. En el nostre cas només ens interessen les referències de tipus *article* i haurem de definir, com a mínim, els camps: *author*, *title*, *journal* i *year*.

- Es considera que el nom d'un autor o editor pot constar de quatre parts diferents: *First*, *von*, *Last*, *Jr.*. Es poden ordenar de diverses maneres, però nosaltres ho farem amb `<von> <last>, <middle>, <first>`. Cal separar múltiples noms amb la paraula `and`.

- L'últim camp d'una referència pot acabar o no amb una coma.

2.3 Característiques

2.4 Disseny del sistema

Hem organitzat el codi del sistema en els mòduls que es llisten a continuació:

- *Raw Content Extraction* (rce): Agrupa totes les classes encarregades d'extreure el contingut dels documents PDF.
- *Information Retrieval* (ir): Encarregat de comunicar-se amb els diferents cercadors disponibles a Internet per obtenir pàgines que contenen informació de la referència que volem extreure.
- *Information Extraction* (ie): Conté tot el codi que permet obtenir la referència a partir d'una pàgina HTML. A més, també és l'encarregat de generar nous *wrappers*.
- *References*: Per una banda fa un anàlisis sintàctic de les referències extretes per poder-les validar. Per l'altra, transforma a BibTeXles referències extretes.
- Base de dades (db): Tal i com indica el seu nom, duu a terme els accessos la base de dades.
- *Main*: Enllaça tots els mòduls anteriors i proporciona punts d'entrada a la interfície d'usuari. Fa de façana del sistema.
- *Graphical User Interface* (gui): Interfície d'usuari més o menys amigable.

La figura 2.1 mostra com interaccionen entre ells.

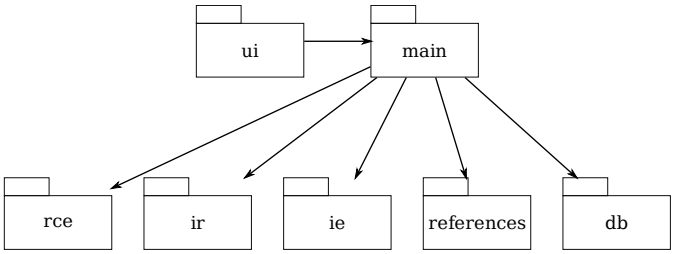

Figura 2.1: Mòduls del sistema

Capítol 3

Cerca de referències

3.1 Extracció dels continguts d'un PDF

El primer pas per aconseguir els objectius proposats és l'extracció del contingut d'un fitxer PDF. Aquest és un dels aspectes que han influït més en l'enfocament que hem donat al sistema, pels motius que es descriuen a continuació.

En un principi, la solució que vam plantejar va ser intentar extreure la referència bibliogràfica d'un document directament del fitxer PDF del qual es disposa. Tot i que a simple vista ja es veu que això pot tenir limitacions (e.g. informació que no es troba dins del text), després de veure com queden els articles al convertir-los a text ens vam allunyar encara més d'aquesta idea. Els llistats 3.1 i 3.2 mostren exemples de les capçaleres de dos articles diferents després d'haver extret el text del fitxer PDF en el que es trobaven. Com es pot veure, no hi ha cap tipus d'estructura que pugui deixar intuir quina part del text correspon a cada fragment d'informació que ens interessa.

```
Characterization and Armstrong Relations for Degenerate
   Multivalued Dependencies Using Formal Concept Analysis
Jaume Baixeries and Jos´ Luis Balc´zar e a
Dept. Llenguatges i Sistemes Inform‘tics, a Universitat
   Polit‘cnica de Catalunya, e c/ Jordi Girona, 1-3, 08034
   Barcelona {jbaixer, balqui}@lsi.upc.es

Abstract. Functional dependencies, a notion originated ...
```

Llistat 3.1: Text corresponent a la capçalera d'un article després d'haver-lo extret d'un PDF

```
2010 Second International Conference on Future Networks

Cloud Computing Research and Development Trend
Shuai Zhang Hebei Polytechnic University College of Science
    Hebei Polytechnic University NO.46 Xinhua West Street
   Tangshan 063009, Hebei Province China zhangshuai@heut.
   edu.cn Xuebin Chen Hebei Polytechnic University College
   of Science Hebei Polytechnic University NO.46 Xinhua
```

```
    West Street Tangshan 063009, Hebei Province China
    chxb@qq.comm
Abstract—With the development of parallel computing,
    distributed [...]
```

Llistat 3.2: Un altre exemple de text extret d'un PDF

3.1.1 Dificultats

Tot hi haver-hi diverses utilitats que permeten l'extracció del contingut d'un fitxer PDF en forma de text pla o HTML, totes presenten problemes similars als que es descriuen a continuació. Les principals dificultats que es troben a l'hora d'obtenir el text són:

- Caràcters especialscom ara Unicode o lligadures (e.g. *fi* es representa com un sol caràcter)
- Sub/Superíndexs: la majoria d'eines els extreuen com un número que forma part de la paraula. Per exemple: $Joan^3$ s'extreu com a *Joan3*
- Flux del text dins del fitxer: Hi ha casos en que el text es troba en diferents columnes i
- Fragmentació de paràgrafs:Relacionat amb el punt anterior. Hi ha ocasions on els paràgrafs es divideixen en un conjunt de línies segons com es troben posicionades dins del document.
- Fitxers protegits dels quals no es pot extreure el contingut
- Documents escanejats:Aquests fitxers només contenen imatge i, no en podem extreure el contingut amb les eines que hem provat.

A banda d'aquestes dificultats tècniques també hi influeix el fet que hi ha un número força reduït de programari lliure que ofereixi aquesta funcionalitat.

3.1.2 Programari

Algunes de les opcions que hem tingut en compte a l'hora d'escollir una llibreria o aplicació d'extracció de text han estat: *PyPDF*, *PDFMiner* o *PDFBox*, tot i que finalment ens hem decantat per *xPDF*.

xPDF és un conjunt d'eines executables des de la línia de comandes que permeten extreure text i altres elements dels fitxers PDF. Es distribueixen sota la llicència GPL v.2 i hi ha binaris tant per Windows com per Linux (que també funcionen per Mac OS). El motiu principal pel qual hem escollit aquesta eina és la qualitat dels resultats. En especial, el fet que no separa els paràgrams en diferents línies i que en la majoria dels casos respecta el flux del text dins del document.

Pel que fa als caràcters especials, transforma bé les lligadures en múltiples caràcters, però té problemes amb la codificació Unicode. Donat que la majoria dels articles científics estan escrits en anglès, aquest és un problema que hem decidit obviar.

3.2 Consultes

El punt més important per poder cercar referències bibliogràfiques a Internet és ser capaços de generar consultes que retornin bons resultats. Una primera idea pot consistir en cercar segons el títol de l'article del qual volem informació. El problema és que bona part dels resultats corresponen a pàgines que fan una referència a aquest article, però que no en donen gaires detalls. Com que a l'extreure el text del PDF la resta de les dades de la capçalera queden desfigurades, és difícil itzar-les per fer les consultes.

Per una altra banda, si intentem fer consultes a partir del contingut del propi article ens trobem amb que en molts casos, els cercadors no el tenen indexat. Una tercera opció, que és la que utilitzem, consisteix en generar les consultes a partir del resum o *abstract* que acompanya a la majoria d'articles i que també acostuma a aparéixer a les pàgines que contenen la referència.

Però com podem saber quina part del text que hem extret correspon al resum? Tot i que en molts articles el primer paràgraf va precedit de la paraula *Abstract*, també n'hi ha molts altres que van precedits d'una paraula completament diferent (e.g. resum o *summary*) o bé per cap. Per tal que el sistema sigui el més general possible, enlloc de fixar-nos en paraules concretes fem servir una expressió regular molt simple que permet trobar cadenes amb un número de paraules determinat.

Un dels trets característics de les capçaleres dels articles una vegada n'hem extret el text és que contenen un nombre elevat de símbols especials. Això ens pot ajudar a distingir entre les parts corresponents a la capçalera i resum. L'expressió regular que obté les consultes és: `([\w()?!]+[ ]){min,max}` i agafarà seqüències de *min* a *max* paraules separades per un espai i formades per caràcters alfanumèrics i un nombre limitat de símbols. Els paràmetres *min* i *max* són configurables. Òbviament, les consultes que ens dóna aquesta expressió no sempre són bones i per tal de contrarrestar aquests errors, en generem un cert nombre que anirem utilitzant mentre no s'obtinguin resultats satisfactoris. De totes maneres, tal i com es pot veure al capítol 6, no és necessari ni generar moltes consultes ni cal que aquestes siguin gaire llargues.

A continuació es llisten cinc consultes extretes d'un article d'exemple. Noteu que les consultes s'envolten de cometes dobles, la forma habitual d'indicar als cercadors que les coincidències han de ser exactes.

- "are known to admit interesting characterizations in terms of Formal"
- "natural extensions of the notion of functional dependency are the"
- "We propose here a new Galois"
- "which gives rise to a formal concept lattice corresponding precisely"
- "o the degenerate multivalued dependencies that hold in the relation"

En molts casos, l'expressió regular anterior també dóna coincidències pel títol de l'article. Per evitar-ho, hem definit un altre paràmetre que defineix el nombre de consultes a saltar-se des del principi de l'article.

3.3 Cercadors

El següent pas després d'haver obtingut un conjunt de consultes és utilitzar-les amb un cercador per tal d'obtenir pàgines amb informació de la referència que volem aconseguir. Al capítol d'introducció, hem esmentat que hi ha cercadors com ara *Google Scholar* o *Microsoft Academic Search* on els resultats només corresponen a publicacions. En un principi, ens va semblar raonable intentar fer ús d'aquests serveis per poder aconseguir els nostres objectius.

El principal problema que hem trobat amb aquests és que no han publicat cap API per tal de permetre les consultes automàtiques des d'aplicacions de tercers. Tot i que hi ha solucions a aquest problema, van en contra dels termes i condicions i els servidors bloquegen massa consultes seguides. Per tant, hem descartat aquesta opció.

Així doncs, ens quedem amb els cercadors habituals i hem preparat la nostra aplicació per tal d'utilitzar les APIs de *Google*, *Yahoo* i *Bing*. Els principals inconvenients són que retornen qualsevol tipus de pàgina i que no tenen indexades algunes blioteques digitals,. Tot i així, també podem aconseguir bons resultats amb l'ús de les consultes adequades.

3.3.1 Ordenació de resultats

La majoria de vegades, no ens convindrà tant l'ordre dels resultats donat pels diferents cercadors sinó que voldrem processar les pàgines per les quals tenim regles d'extracció d'informació. És per això que

3.3.2 Altres Ajustaments

Depenent de ldel tipus de fitxers dels que disposem la qualitat dels resultats obtinguts amb els cercadors poden variar considerablement. Això suposa la necessitat d'ajustar alguns paràmetres per tal de poder adaptar el sistema a l'ús de cadascú. A la secció sobre la generació de consultes (3.2), ja hem comentat la possibilitat d'ajustar el mínim i màxim de termes a cercar, però hi ha altres opcions que es poden configurar.

En algunes ocasions, es dóna el cas que la consulta generada no és prou restrictiva, ja sigui perquè no és prou llarga o bé perquè està formada per paraules molt generals. Al cercar amb aquestes consultes s'obté una llarga llista de resultats, la majoria dels quals no tenen res a veure amb la informació que estem buscant. Per contrarestar-ho, hi ha la possibilitat d'indicar al sistema que ometi els resultats i provi amb la següent consulta. A l'hora d'assignar el valor d'aquest paràmetre, també s'haurà de tenir en compte el tipus d'articles dels que es vol informació. Per exemple, articles populars tindran un número de coincidències rellevants gran i, per tant, haurem d'assignar un valor relativament alt, ja que un valor baix farà que descartem resultats bons. En canvi, per articles poc corrents, ens interessarà el contrari.

Per una altra banda, hi ha ocasions en que els cercadors tenen tendència a retornar resultats que, tot i coincidir amb la consulta que li hem donat, corresponen a una pàgina que no ens aporta massa informació. Per tal d'ajudar a l'aplicació a descartar resultats dolents, podem indicar-li pàgines que volem ometre a partir d'una llista negra. Per exemple, sabem que les pàgines sobre els autors de la biblioteca digital *ACM Portal* contenen un llistat de tots els articles d'un mateix autor, però que no contenen suficient informació com per extreure referències. En aquest cas

voldrem descartar els resultats que comencen per http://portal.acm.org/author_page.cfm?id=id-autor.

3.4 *Multithreading*

Un dels inconvenients més grans que implica el fet d'haver d'accedir a Internet, és que el temps perdut esperant dades és molt alt. Per reduir-lo, s'ha estudiat la possibilitat d'utilitzar diferents fils d'execució per fer més d'una consulta de forma més o menys simultània. La taula següent mostra una comparativa del temps necessari per obtenir mútliples pàgines web de forma seqüencial o bé utilitzant fins a cinc fils d'execució diferents. Les pàgines pàgines corresponen a consultes aleatòries a *Google* per evitar l'efecte dels *proxies* i *caches*.

2 pàgines		5 pàgines		10 pàgines		20 pàgines	
Seq.	5 Threads	Seq.	5 Threads	Seq.	5 Threads	Seq.	5 Threads
0.9010	0.5481	2.1830	0.6612	4.3153	1.5914	7.9295	2.5949
0.7467	0.3795	2.1558	0.7441	4.3186	1.2311	8.5483	2.1958
0.7678	0.5641	2.0645	0.5383	9.2930	1.4415	8.7202	2.5749
0.7421	0.3876	2.0684	0.8551	4.9859	1.5294	8.4732	2.2841
0.9674	0.5477	2.1510	0.8550	5.3600	1.3116	9.2901	2.2257
Mitjana:							
0.8250	0.4854	2.1246	0.7307	5.6546	1.4210	8.5923	2.3751
Guany:							
-44.96%		**-65.6%**		**-74.87%**		**-72.35%**	

Tot i que auqestes proves no siguin gaire riguroses, són suficients per poder-nos fer una idea força clara sobre la millora que s'obté utilitzant múltiples fils respecte no fer-ho.

Sobre la forma d'implementar-ho, hem creat un *pool* amb un número màxim configurable de fils d'execució que es van reutilitzant mentre queden referències per extreure. Bàsicament, tenim una cua amb les rutes als fitxers PDF i una cua de sortida amb el resultat d'extreure les referències. Cada *thread* va processant fitxers de la cua d'entrada mentre aquesta no és buida. El número de fils màxim dependrà del tipus de connexió del que es disposi.

Capítol 4

Extracció de referències

En aquest capítol tractarem

En el nostre context, anomenarem *wrapper* a un troç de codi que podem utilitzar per extreure una peça d'informació concreta d'un document.

Podem imaginar-ho com filtre que només ens deixa veure una part del document que ens interessa.

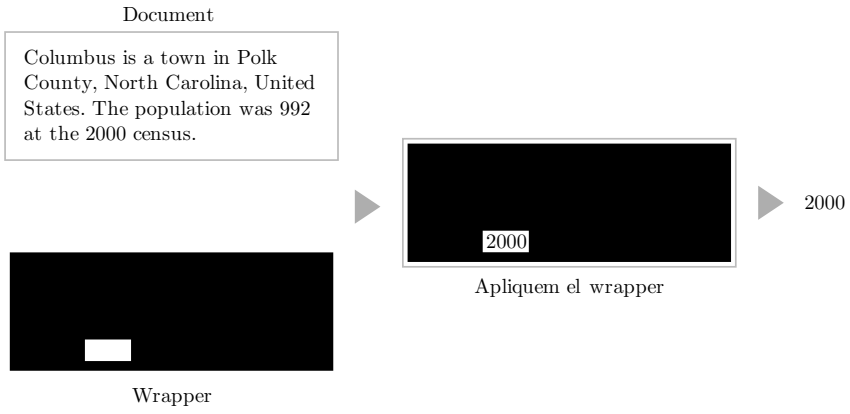

Figura 4.1: Funció d'un *wrapper*

4.1 *Wrappers* a mà

4.2 Inducció de *wrappers*

4.2.1 Generació automàtica de regles

Ruta d'un element HTML

Expressió regular

4.2.2 Avaluació dels *wrappers*

Una vegada hem generat el conjunt dels *wrappers* possibles per a un conjunt de documents, cal que avaluem quins d'ells funcionen millor. Utilitzem un sistema de vots positius i negatius i en calculem la mitjana amb la següent fórmula:

$$score = \frac{vots\ positius}{vots\ totals}$$

4.2.3 Reaprenentatge

El sistema està dissenyat per tal que, quan hi ha una davallada en el nombre de referències extretes correctament, provi de reaprendre els *wrappers* automàticament a partir dels exemples que té emmagatzemats d'execucions passades.

Capítol 5

Inducció de *Wrappers*

Capítol 6

Anàlisi de resultats

En aquest capítol es mostren les principals proves realitzades per cadascuna de les tres parts del sistema que hem descrit als capítols anteriors

6.1 Cerca de referències

En primer lloc provarem com de bé ho fa el sistema a l'hora de cercar pàgines a Internet que continguin informació sobre un article concret. Els tests que hem dut a terme consisteixen en:

1. Obtenir una sèrie de consultes d'un llistat de documents PDF
2. Cercar cadascuna de les consultes amb: *Google*, *Bing* i *Yahoo*
3. Per cadascun dels resultats obtinguts, analitzem si és bo o no
4. Comptabilitzem el número de consultes que han fet falta per obtenir el primer *bon* resultat

Sobre els passos anteriors, hem de notar que per tal classificar els resultats en bons i dolents nomes comprovem si part de la informació que volem es troba dins de la pàgina resultant. Aquesta no és una solució perfecta, però ens permet fer una aproximació sobre la quantitat de fitxers pels quals en podem trobar la referència.

Una altra qüestió sobre la implementació dels tests, és que els resultats obtinguts se solen repetir entre consultes del mateix article. Per estalviar temps i evitar fer moltes peticions seguides als mateixos servidors (que podrien resultar en un bloqueig), deixem uns segons entre petició i petició i emmagatzemem cada resultat de manera que només l'haguem de demanar una sola vegada.

Tambe hem de comentar que en molts casos, alguns dels resultats corresponen al mateix fitxer PDF del qual estem buscant informació i que, per tant, els hem d'ometre.

6.2 Extracció de referències

6.3 Inducció de *wrappers*

Capítol 7

Conclusions i Treball Futur

7.1 Objectius Assolits

7.2 Possibles Millores

Bibliografia

[GBL98] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. Citeseer: an automatic citation indexing system. In *INTERNATIONAL CONFERENCE ON DIGITAL LIBRARIES*, pages 89–98. ACM Press, 1998.

Apèndix A

Extracció Contingut PDF

Apèndix B

Resultats dels tests

A continuació es mostren els resultats complets de totes les proves realitzades a la nostra aplicació. L'explicació d'aquests números s'explica al capítol 6.

Apèndix C

Biblioteques utilitzades

A continuació es llisten les diferents biblioteques i mòdules *Python* que s'han utilitzat en l'aplicació

- SimpleJSON
- DiffLib
-