BibTeX Bibliography Index Maker: Informe del Projecte

Ramon Xuriguera
rxuriguera@gmail.com

Març 2010

1 Objectius del projecte

BibTeX Index Maker és una eina d'ajuda a la creació d'índexos bibliogràfics pensada com un complement a aplicacions de maneig de referències ja existents com poden ser *JabRef*[1] o *Mendeley*[2].

La principal funcionalitat 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.

2 Treball existent

Actualment existeixen nombroses aplicacions dedicades al maneig de referències. Algunes d'elles utilitzen les metadades dels fitxers per tal de trobar informació com ara el títol o l'autor, però cap de les eines que hem trobat llegeix el contingut dels documents.

Empreses com ara *Google* o *Microsoft* agafen la informació de documents PDF per oferir serveis com ara *Scholar* o *Academic Search*, però no ofereixen el codi font i per tant, no sabem com funcionen. *CiteSeer* és un projecte *open source* de característiques similars, però que també té limitacions. El sitema funciona analitzant la bibliografia dels articles, però té problemes per obtenir els camps de la capçalera del propi fitxer, que és el que ens interessa.

3 Descripció detallada

Per entendre el per què de l'enfocament que hem donat a la nostra aplicació, cal que comencem explicant el principals problemes que que hem trobat: l'extracció de text dels fitxers PDF. Aquest procés no és trivial i, per tant, les eines que que es poden trobar per fer-ho no obtenen resultats gaire bons. En especial hi ha problemes amb el flux del text dins el fitxer, els caràcters Unicode, les lligadures (e.g. *fi* es representa amb un sol caràcter), superíndexos, etc. Als articles científics, la informació que volem extreure acostuma a contenir molts caràcters d'aquest tipus.

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

```
Enhancing Prediction on Non-dedicated Clusters
Joseph Ll. L rida1 , F. Solsona1 , F. Gin 1 , J.R. Garca2 ,
e e i a M. Hanzich2 , and P. Hern ndez2 1
...
```

Llistat 1: Part del text extret d'un article

A més, els documents no acostumen a contenir certa informació necessària per poder crear la referència (e.g. pàgines, editor) i la seva estructura sol ser molt variada depenent de la revista en la qual s'ha publicat l'article. Aquestes dificultats ens han fet mirar d'abordar el problema d'alguna altra manera.

La solució que hem trobat és fer ús de les biblioteques digitals disponibles a Internet, per exemple *SpringerLink* o *ACM*. Per poder aconseguir la referència corresponent a un article seguirem el procediment que mostra la figura 1. Bàsicament, per cada document PDF, l'aplicació:

1. Extreu el contingut en forma de text.
2. Genera un llistat de consultes a partir del text. Agafa frases de l'*abstract*.
3. Obté els resultats d'aquestes consultes en algun cercador (Google, Bing, etc.).
4. Mira d'extreure la referència de les pàgines retornades pel cercador.
5. Comprova que les dades de la referència corresponen al fitxer.

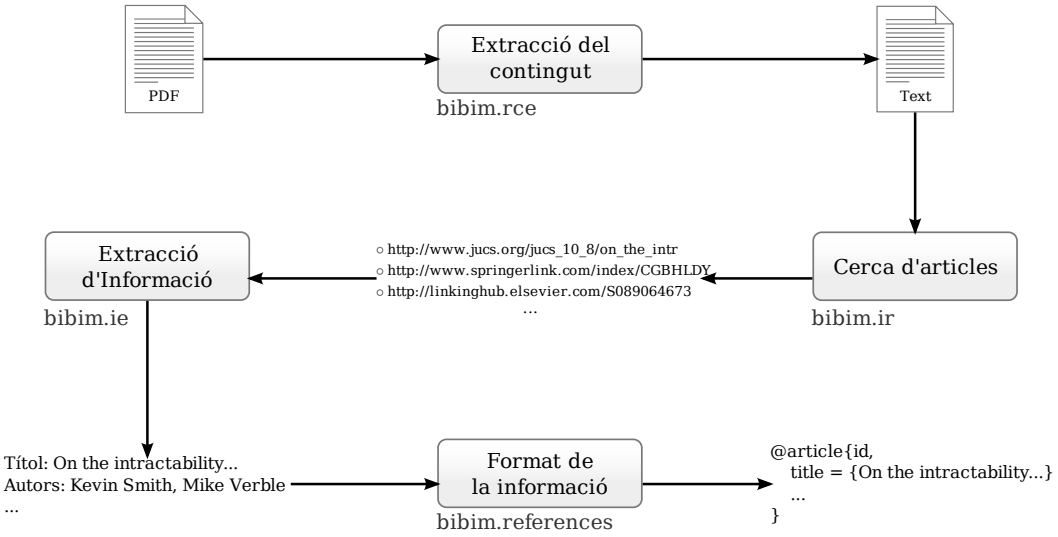

Figura 1: Procediment per obtenir la referència a un article

3.1 Extracció de contingut

El pas més interessant d'aquest procés és l'extracció de la referència a partir de les pàgines resultants de fer la cerca. Com és lògic, els resultats obtinguts d'Internet corresponen a pàgines completament diferents i el nostre programa ha de ser capaç de detectar on es troba cada fragment d'informació i agafar-lo. A primera vista podria semblar que si les pàgines són tant diferents, el fet de buscar l'article a Internet no ens ha solucionat res. La principal diferència, però, és que la informació d'aquestes pàgines està estructurada.

Una primera idea per fer-ho és tenir un conjunt de *wrappers* de manera que cadascun d'ells sap localitzar la informació en una de les pàgines. Aquesta solució no és perfecta, però pot arribar a funcionar. Hem pogut comprovar que els articles d'un mateix camp de coneixement acostumen a aparéixer a les mateixes biblioteques. Això implica que, a la pràctica, amb un número no molt gran de *wrappers* es poden obenir resultats prou bons. El principal problema d'aquesta tècnica és que quan alguna de les pàgines de les quals es depèn canvia la seva estructura, cal modificar també el *wrapper* corresponent.

Una alternativa és la possibilitat de generar aquests *wrappers* de forma atomàtica a partir d'un número reduït d'exemples. Això no només facilita la possibilitat d'extreure informació de noves pàgines sinó que quan l'aplicació detecta que els resultats comencen a empitjorar es poden reaprendre les regles. Per poder fer això, necessitem tenir les dades de les extraccions anteriors emmagatzemades a una base de dades d'exemples. Per cada exemple hem de tenir, com a mínim, la URL de la pàgina i les dades que volem obtenir. El procediment és el següent:

1. Cerquem cada un dels camps que volem extreure dins de la pàgina.
2. Establim la posició de l'etiqueta HTML que conté la informació dins de la pàgina.
3. Trobem la posició del tros d'informació dins de l'etiqueta HTML
4. Generem la regla d'extracció
5. Validem la regla amb la resta d'exemples

3.2 Mòduls de l'aplicació

Hem decidit dividir el codi de l'aplicació en diferents mòduls segons la seva funcionalitat, tot mirant que entre ells hi hagi el mínim de dependències possibles. El resultat es pot veure a continuació. A la figura 1 de la secció anterior, ja hem pogut veure què fa cadascun d'ells, excepte `bibim.ui`, que correspon a la interfície d'usuari i `bibim.db`, que conté el codi relacionat amb la base de dades.

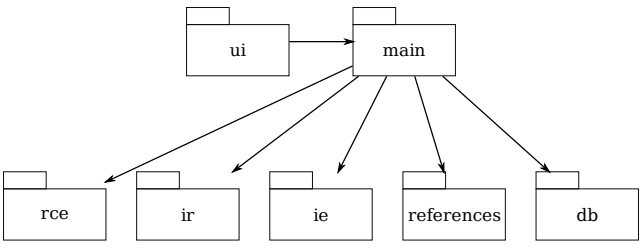

Figura 2: Diagrama de mòduls i dependències

4 Paquets de treball

La taula següent mostra el llistat de paquets de treball acompanyats del temps estimat que ha calgut o caldrà per finalitzar-los i l'estat en què es troben. Les tasques amb un estat $\geq 95\%$ es consideren tasques completes, però susceptibles de correccions.

Tasca	**Temps estimat**	**Estat**
Recerca i proves de concepte	20 dies	100%
Infraestructura (repositori, *build tracker*, etc.)	4 dies	100%
Extracció del text dels PDF (`bibim.rce`)	4 dies	95%
Obtenció de pàgines amb la referència (`bibim.ir`)	9 dies	93%
Obtenir consultes del text	1 dia	95%
Browser per obtenir consultes	2 dies	95%
Cerca de les consultes (múltiples cercadors)	5 dies	95%
Ordenar resultats de la cerca	1 dia	80%
Extracció d'informació (`bibim.ie`)	21 dies	**26%**
Wrappers a mà	5 dies	60%
Generació automàtica de *wrappers*	12 dies	13%
Extracció d'exemples	1 dia	50%
Generació de regles	12 dies	10%
Wrapper genèric	3 dies	15%
Tractament de referències (`bibim.references`)	5 dies	94%
Parsing	2 dies	95%
Generació	2 dies	95%
Crear índexs	1 dia	90%
Base de dades (`bibim.db`)	3 dies	95%
Mòdul principal (`bibim.main`)	8 dies	93%
Escaneig de fitxers	1 dia	95%
Codi principal de l'aplicació	3 dies	95%
Threading	2 dies	95%
Validació de referències	2 dies	85%
Interfície d'usuari (`bibim.ui`)	4 dies	**15%**
Memòria	-	$\approx$ **10%**

Alguns aspectes que cal comentar:

- La implementació de *wrappers* a mà està aturada. Aquests *wrappers* seran susbtituïts si s'obtenen bons resultats amb la generació automàtica.
- El *parsing* de referències és necessari per poder validar els camps de la referència en aquells casos en que se n'extreu el codi complet BibTeX.
- Els *threads* s'utilitzen per poder processar més dun fitxer a la vegada. Per exemple, mentre s'estan esperant els resultats d'una cerca a Internet, es pot anar extraient el text d'un altre PDF.