Títol: Entorn per a la inducció de patrons d'extracció, orientat a la Wikipedia

Volum:I/I

Alumne:Jaume Martí Farriol

Director/Ponent:Horacio Rodríguez Hontoria

Departament:Llenguatges i Sistemes Informàtics

DADES DEL PROJECTE

Títol del Projecte: Entorn per a la inducció de patrons d'extracció, orientat a la Wikipedia

Nom de l'estudiant: Jaume Martí Farriol
Titulació: Enginyeria Informàtica
Crèdits: 37.5
Director/Ponent: Horacio Rodríguez Hontoria
Departament: Llenguatges i Sistemes Informàtics

MEMBRES DEL TRIBUNAL *(nom i signatura)*

President:

Vocal:

Secretari:

QUALIFICACIÓ

Qualificació numèrica:
Qualificació descriptiva:

Data:

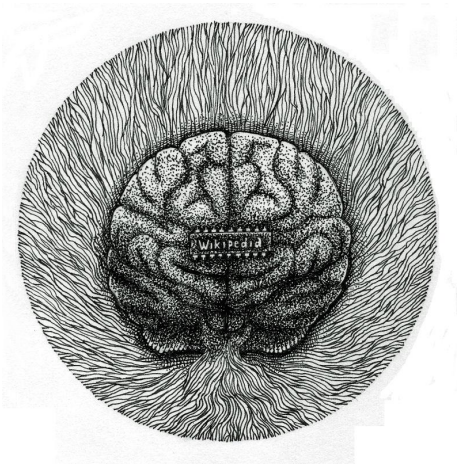
JAUME MARTÍ

ENTORN PER A LA INDUCCIÓ DE PATRONS D'EXTRACCIÓ ORIENTAT A LA WIKIPEDIA

[2 de novembre de 2009 at 19:14]

ENTORN PER A LA INDUCCIÓ DE PATRONS D'EXTRACCIÓ ORIENTAT A LA WIKIPEDIA

JAUME MARTÍ



Enginyeria Informàtica

Llenguatges i Sistemes Informàtics
Setembre 2009

[2 de novembre de 2009 at 19:14]

Jaume Martí: *Entorn per a la inducció de patrons d'extracció orientat a la Wikipedia,* Enginyeria Informàtica, © Setembre 2009

ABSTRACT

This report describes a system that takes advantage of Wikipedia's structure, that combines structured data with non structured data, to autonomously induce extraction patterns from entities which correspond to the value of a non taxonomic relation between this, and another entity. The relation and one of the entities is defined by an Infobox, which are structured data contained in some articles.

RESUM

Aquesta memòria descriu un sistema que aprofita l'estructura de la Wikipedia, que combina dades estructurades amb dades no estructurades, per induïr de forma completament automàtica, patrons per a l'extracció d'entitats que corresponen al valor d'una relació no taxonòmica sobre una altre entitat. La relació i una de les entitats, venen definits per les Infobox, que son dades estructurades contingudes en alguns dels articles.

ÍNDEX

LLISTA DE FIGURES

LLISTA DE TAULES

ACRÒNIMS

SPARQL SPARQL Protocol and RDF Query Language

SQL Structured Query Language

RDF Resource Description Framework

XML Extensible Markup Language

EN Entitat Nombrada

Part I

INTRODUCCIÓ

1 PRESENTACIÓ

El projecte descrit en aquesta memòria, te com a propòsit la mineria de text en l'àmbit de la Wikipedia. Usant la meta-informació d'aquesta, es pot generar de forma automàtica les dades necessàries per realitzar la inducció de patrons d'extracció.

La mineria de text consisteix en el procés que te com a objectiu derivar informació útil d'un text escrit en llenguatge natural. Aquestes tasques solen consistir en extracció d'entitats, modelatge de relacions entre entitats, categorització de textos, resum de textos, entre d'altes.

Concretament aquest projecte es troba dins de l'àrea de l'extracció d'informació, que es una forma de mineria de text que consisteix en la localització de peces de text rellevants dins d'un bloc de text natural, per tant, el procés consisteix en extreure informació estructurada d'una font no estructurada.

El tipus d'informació que es pot extreure es pot classificar en informació taxonòmica i informació relacional. La informació taxonòmica consisteix en aquella que correspon a una relació taxonòmica ("es un", "es part de", etc) entre dues entitats. Per altre banda la informació relacional correspon a la resta de relacions entre entitats (p.e. "en Pere té 43 anys", correspondria a la relació "edat" entre "Pere" i 43).

La informació que s'extreu consisteix en entitats relacionades, no taxonòmicament, amb una altre entitat. On la relació i una de les entitats venen donades, i el que es pretén es extreure l'altra entitat. Aquestes entitats corresponen a elements atòmics dins d'un text, que no pertanyen a elements propis del llenguatge (p.e. verbs, adjectius, etc.) constituïts per blocs de text alfanumèrics. En el context d'aquest projecte, es pretén extreure entitats nombrades, dates i valors numèrics del corpus de la Wikipedia, on aquestes entitats corresponen als valors d'atributs de les Infoboxes.

Les entitats nombrades son conjunts de paraules, nombres, i demés tokens textuals que fan referència a una entitat concreta (p.e. un nom de persona, el nom d'un gen, etc), i que comparteixen alguna característica i per tant, se'ls pot catalogar (p.e. E.N. de persones, d'organitzacions , d'indrets geogràfics, etc).

El procés d'extracció d'informació es realitza en dos passos. En un primer pas es generen un conjunt de patrons d'extracció, que operen sobre un espai de característiques que pretén abstraure, i simplificar l'estructura del text natural sobre el que es treballa. Aquests patrons son induïts usant algoritmes d'aprenentatge automàtic.

La meta informació usada en aquest projecte es la pròpia categorització del corpus de la Wikipedia en articles, útil per relacionar una entitat nombrada amb un corpus que conté informació sobre aquesta. També s'usa un tipus de Template (també anomenat Patró) les instàncies del qual s'anomenen Infobox (també anomenat Fitxa), de les quals un article en pot contenir de diversos tipus; i conté fets sobre aquesta entitat nombrada, en forma de termes (atribut, valor). La Wikipedia conté molta més meta-informació útil per extreure fets sobre una entitat nombrada (p.e. categories d'articles, vincles...), però aquest projecte no en fa ús.

El projecte, s'havia de limitar en algun punt, degut a que s'havia d'acabar en un període limitat de temps. Per tant s'ha optat per extreure únicament els fets definits en les Infoboxs, i presentar el resultat en forma de graf en el format Resource Description Framework (RDF), sense usar cap estructura ontològica.

Per tant, el projecte consisteix en un entorn adaptable, per a la inducció de patrons d'extracció sobre entitats nombrades corresponents a fets, definits en Infoboxs (es a dir, atributs), i finalment l'extracció d'aquestes entitats nombrades i la seva posterior estructuració en forma de graf. El sistema també proporciona mètriques per tal d'avaluar el rendiment del procés d'extracció d'un atribut concret, i un interpret de SPARQL Protocol and RDF Query Language (SPARQL) per a realitzar consultes sobre els resultats.

Per a demostrar el correcte funcionament del sistema, i per il·lustrar el seu ús, es realitza l'extracció de la Entitat Nombrada (EN) "capital", definida en molts tipus d'Infobox, com ara en l'infobox "Ficha de país" i també de la data de naixement. L'extracció d'aquests dos atributs presenta problemes molt diferents, d'aquesta manera es pot veure com es comporta l'entorn en diferents contextos.

La Wikipedia actualment, es un corpus d'informació enorme, que en el cas de l'Anglès per exemple, ocupa 150GB d'informació només en text. Per tant, per a poder aprofitar tota aquesta font d'informació també es necessita dissenyar un sistema escalable per al preprocessament i l'emmagatzematge/consulta de la informació.

1.1 ANTECEDENTS EN LA MINERIA SOBRE EL CORPUS DE LA WIKIPEDIA

- DBPedia: Es un projecte que pretén extreure coneixement directament de les Infobox de la Wikipedia, i generar un conjunt de triples, en format RDF, com a resultat. El projecte usa una ontologia, però no es induïda per meta-informació de la Wikipedia, sinó que es va dissenyar a part. El codi està disponible al públic.
- Kylin [3]: Es un projecte que intenta completar, o crear, Infoboxs en articles de la Wikipedia, usant algoritmes d'aprenentatge automàtic. El projecte descrit en aquesta memòria, usa Kylin com a referent.

1.2 MOTIVACIÓ

Actualment els processos d'extracció d'informació es classifiquen en dos grans grups. Per una banda hi ha els que es basen en el coneixement, es a dir, s'usa el coneixement humà adquirit en l'àrea sobre la que han de treballar, per dissenyar-los. I l'altre es el basat en aprenentatge automàtic supervisat, que s'usa per induir un model que permet l'extracció de la informació desitjada. El problema dels primers es que requereixen un procés molt costos, tant a nivell de temps com a nivell econòmic; a més a més, aquesta tècnica produeix un sistema que només es aplicable en un domini molt específic. Pel que fa a l'ús de tècniques d'aprenentatge automàtic, requereix d'un extens corpus etiquetat manualment, per tant, no eliminen la intervenció humana. Amb aquest projecte s'aprofita el contingut semi-estructurat de la Wikipedia, concretament les Infobox, per a automatitzar completament el procés d'inducció de patrons d'ex-

tracció, generant els exemples positius i negatius necessaris pel procés d'aprenentatge automàtic.

1.3 OBJECTIUS

El propòsit d'aquest projecte es aprofitar la naturalesa de la Wikipedia, una font d'informació de dimensions considerables, que combina dades semi-estructurades i dades no estructurades; per automatitzar el procés d'aprenentatge supervisat d'un sistema d'extracció d'informació, en el domini de l'extracció de fets descrits en les Infoboxes de la Wikipedia.

1.4 ORGANITZACIÓ D'AQUESTA MEMÒRIA

- El capítol 2 conté una anàlisi de l'estructura de la Wikipedia, i de les diverses vessants que es poden enfocar per extreure'n informació; aprofundint en aquelles que son més rellevants per a aquest projecte. Aquest capítol es basa principalment en l'article [7], i s'ha adaptat a aquest projecte, i en la mesura del possible, a la versió castellana de la Wikipedia.
- El capítol 3 descriu l'arquitectura general del software desenvolupat per a aquest projecte.
- El capítol 4 descriu tots els recursos de software que s'han usat, externs a aquest projecte.
- Per poder extreure informació de la Wikipedia, primer s'ha de desenvolupar un primer component que processi el llenguatge de marques, Wiki, dels articles de la Wikipedia, l'estructuri, i l'emmagatzemi de forma que sigui accessible de forma escalable independentment al volum del corpus. El capítol 5 descriu l'entorn que s'ha desenvolupat per a l'accés estructurat al corpus de la Wikipedia.
- El capítol 6 descriu el procés d'aprenentatge.
- El capítol 7 descriu el procés d'extracció de la informació.
- El capítol 8 descriu les mètriques que s'usen per avaluar el rendiment, i aquestes s'usen per avaluar l'exemple proposat en el capítol 9.
- El capítol 9 presenta un conjunt de casos d'avaluació, en els que s'extreu un atribut, i s'avalua el rendiment de l'entorn en el procés d'extracció.
- El capítol 10 conté la planificació seguida en el desenvolupament d'aquest projecte.
- El capítol 11 conté les conclusions extretes de la realització d'aquest projecte.
- L'apèndix conté el manual d'ús del software desenvolupat per a aquest projecte i la descripció de les etiquetes morfosintàctiques usades.

2 ESTRUCTURA DE LA WIKIPEDIA

Aquest capítol es una descripció de l'estructura de la Wikipedia, i principalment de les parts d'aquesta estructura que s'usen en aquest projecte. El capítol està basat en l'article [7] i ha estat adaptat per a aquest projecte.

La Wikipedia es una enciclopèdia multilingüe, amb 271 llenguatges, i que es confecciona a través de la metodologia anomenada CCC (Collaborative Content Creation) on qualsevol persona pot col·laborar.

Les enciclopèdies tradicionals estan organitzades en articles ordenats alfabèticament, amb referències 'creuades' a altres articles, refèrencies externes a literatura acadèmica i un index general de temes. Aquesta estructura s'ha mantingut, encara que adaptada per a ser presentada en un entorn web; i a més s'han afegit capacitats com un motor de cerca per accedir als articles a traves de paraules claus contingudes en el títol de l'article o en el seu cos, etc.

S'usarà la Wikipedia en la seva versió en Castellà, i en menor mesura en Anglès, per a il·lustrar els conceptes descrits en aquest capítol.

2.1 ARTICLE

La unitat principal d'informació en la Wikipedia es l'article. L'enciclopèdia conté més de 10 milions d'articles en les diferents llengües que suporta, La versió anglesa conté 2,4 milions d'articles, sense contar pàgines de redirecció i de desambiguació. Uns 1,8 milions son articles amb almenys un vincle, i 30 paraules o més de text en el seu cos. Els articles estan escrits en text lliure, combinat amb un llenguatge de marques que permet estructurar el text, afegir vincles, o instanciar plantilles. El text d'un article ha de seguir un estil editorial i estructural definit en el Manual of Style, encara que no tots els articles el segueixen. Principalment el que s'exigeix es:

- Cada article ha de descriure un únic concepte, i dos articles no poden descriure el mateix concepte.
- Els títols dels articles son frases curtes, similars a les entrades d'un diccionari.
- Els termes similars son redirigits a un article usant pàgines de redirecció.
- Les pàgines de desambiguació presenten els possibles significats d'un terme per a que es seleccioni l'article que es desitgi.
- Els articles comencen amb una breu descripció del tema sobre el que versa l'article, i la primera frase defineix el concepte i el seu tipus.
- Els articles contenen vincles que els relacionen amb altres articles.

La primera frase defineix de forma genèrica el concepte i especifica el tipus. El títol de l'article solen ser dues paraules, però habitualment es sol afegir entre parèntesis un terme que el diferencii d'altres conceptes

Propulsor iónico

(Redirigido desde Propulsor ionico)

Un **propulsor iónico** o **motor iónico** es uno de los distintos tipos de propulsión espacial, específicamente del tipo eléctrica. Se utiliza un haz de iones (moléculas o átomo con carga eléctrica) para la propulsión. El método preciso para acelerar los iones puede variar, pero todos los diseños usan la ventaja de la relación carga-masa de los iones para acelerarlos a velocidades muy altas utilizando un campo eléctrico. Gracias a esto, los propulsores iónicos pueden alcanzar un impulso específico alto, reduciendo la cantidad de masa necesaria, pero incrementando la cantidad de potencia necesaria comparado con los cohetes convencionales. Los motores iónicos pueden desarrollar una orden de magnitud mayor de eficacia de combustible que los motores de cohete de combustible líquido, pero restringidos a aceleraciones muy bajas por la relación potencia-masa de los sistemas disponibles.

Prueba de un propulsor iónico

El principio del propulsor iónico data de los conceptos desarrollados por el físico Hermann Oberth y su obra publicada en 1929, *Die Rakete zu den Planetenräumen*. El primer tipo de motor iónico, conocido como propulsor iónico de tipo Kaufman, se desarrolló en los años 1960 por Harold R. Kaufman, trabajando para la NASA y basados en el Duoplasmatrón.

Figura 1: Pàgina de l'article "Propulsor iónico"

amb el mateix nom, com ara Júpiter_(planeta) Júpiter_(mitología). La Wikipedia distingeix entre majúscules i minúscules en els títols dels articles a excepció de la primera lletra. per exemple "Optic Nerve (còmic)" Optic nerve (Anatomia).

2.2 PÀGINES DE DESAMBIGUACIÓ

En una pàgina de desambiguació apareixen llistats un conjunt d'articles amb títols iguals o molt similars, seguits d'una breu descripció de cada un. Això fa de les pàgines de desambiguació un recurs atractiu com a font d'homònims. El motor de cerca de la Wikipedia si no pot trobar un article que concordi exactament amb la cerca, pot portar a l'usuari a una pàgina de desambiguació, on l'usuari elegeix entre un conjunt d'articles amb el mateix títol, o títols similars. Aquestes pàgines es creen usant certs patrons, i es poden identificar en funció de si estan assignades a una categoria concreta, instancien certs patrons per a la creació de pàgines de desambiguació, o ho indica en el títol que es una pàgina de desambiguació.

2.3 PÀGINES DE REDIRECCIÓ

Les pàgines de redirecció no contenen text exceptuant una directiva que indica a quin article s'ha de redirigir la pàgina. La versió anglesa de la Wikipedia consta de 3 milions de pàgines de redirecció. Aquestes pàgines solen ser plurals (Bibliotecas), tecnicismes, errors comuns o altres variants (Propulsión iónica). La idea es tenir un únic article per concepte, amb el títol desitjat, i crear redireccions des de termes equivalents. Això facilita la mineria ja que ens estalvia haver de resoldre sinònims.

2.4 HYPERVINCLES

L'ús dels hypervincles està molt extés en la Wikipedia, només en la versió anglesa ja n'hi ha 60 milions, amb 25 per article de mitjana. A l'igual que en la resta de pàgines web, els vincles proporcionen informació addicional sobre els temes discutits en els articles, i permeten descobrir informació interessant que no estaves buscant. Els vincles poden dirigir-te a una altre part de la mateixa pàgina, a una pàgina diferent de la mateixa versió de la Wikipedia, a una Wikipedia d'un altre llenguatge o a una altre pàgina Web. Com que els vincles normalment estan representats per paraules diferents al títol de l'article, també son una font útil de sinònims. A més a més, a vegades un mateix conjunt de termes s'usa per descriure vincles a articles diferents, per tant també es pot usar com a font de termes polisèmics. També es pot usar per determinar el significat més probable d'un terme en funció de la proporció de vincles d'un cert terme que van a cada article. També es pot usar la xarxa de vincles entre articles per determinar la relació entre articles, ja que un vincle entre dos articles implica que estan relacionats d'alguna manera.

2.5 CATEGORIES

Els col·laboradors de la Wikipedia son encoratjats a que assignin categories als articles. Per exemple l'article exemple està contingut en la categoria "Astronáutica". Les categories estan organitzades en una jerarquia (taxonòmicament), d'aquesta manera la categoria "Astronáutica" conté, per exemple, les categories "Exploración espacial", i "Astrodinámica". En la versió anglesa de la Wikipedia hi han casi 400.000 categories, amb una mitja de 19 articles i 2 subcategories cada una. Les categories no son articles, encara que si que son pàgines, i s'usen com a nodes per organitzar els articles amb un mínim de text explicatori, que en aproximadament un terç dels casos corresponen a conceptes que necessiten més explicació. En aquests casos estan aparellats amb un article del mateix nom. La categoria "Medalla Fields" està aparellada amb l'article "Medalla Fields" i conté la llista de galardonats amb aquesta medalla. Altres no estan aparellades amb cap article i serveixen únicament per organitzar el contingut a través d'alguna característica en comú entre un conjunt d'articles, per exemple la categoria "Lagos por superficie" conté els llacs del mon amb una superfície superior als 3000 km2. L'estructura de categories no te forma d'arbre sinó més bé de graf dirigit, que conté algun cicle. Per norma general els cicles estan desencoratjats, tot i que hi ha certs casos que es considera acceptable; com per exemple en la Wikipedia anglesa la categoria "education" esta dins de "social sciences" que alhora esta dins de "academic discipline" i aquesta última està dins de "education"; aquest cicle del graf es correcte ja que descriu l'educació consistent en educar a altres persones. L'estructura de categories de la Wikipedia es una característica relativament novedosa i menys visible que els articles; per tant rep considerablement menys atenció que els articles i per tant conté més errors, inconsistències, redundàncies, etc. Els vincles entre categories poden representar tot tipus de relacions encara que principalment representen una relació de pertenencia a una classe, encara que també poden representar una relació "part de"(contingut en), entre d'altres. Això no ha impedit als investigadors de realitzar mineria sobre les categories de la Wikipedia.

Figura 2: Atributs amb els que el software pot treballar

2.6 PLANTILLES I INFOBOXES

Les plantilles son un sistema que permet reaprofitar un bloc de text estructurat i adaptar-lo a diferents contextos usant paràmetres. La Wikipedia conté uns 174.000 patrons, els quals han estat invocats 23 milions de cops. Els patrons s'usen entre d'altres coses per marcar articles que necessiten atenció, ja sigui perquè son imparcials, estan mal escrits, els hi falten referències, etc. També poden servir per indicar certs tipus de pàgines, o per facilitar la construcció d'aquestes, com per exemple en el cas de pàgines destacades per la seva qualitat, o pàgines de desambiguació. Una altra aplicació típica es per indicar que el títol de l'article te homònims, i proporciona el vincle a la pàgina de desambiguació corresponent. Una infobox es un tipus especial de patró que conté informació factual sobre l'article en un format estructurat en parelles d'atribut, amb el seu corresponent valor.

En la Wikipedia anglesa hi ha unes 3000 infoboxes que van des de descripcions d'especies vegetals a estratègies descrivint com iniciar partides d'escacs. A mode d'il·lustració, la FIG mostra l'infobox "Ficha de artista" del pintor Pablo Picasso. Aquesta infobox conté fets com ara el nom real de l'artista, o les seves obres destacades; i per ser inclosa

en l'article, només s'ha hagut d'instanciar l'infobox, passant-li com a paràmetres els valors dels atributs que es mostren. Les infobox resulten especialment atractives, ja que son una font d'informació estructurada sobre una entitat (la que descriu l'article), aquesta informació poden ser relacions amb altres entitats, com ara les obres d'un artista, o valors atòmics sobre aquesta entitat, com ara l'edat. Tot i això les infobox estan pensades per ser llegides per persones i han estat creades per autors amb diferents criteris, amb les implicacions que això comporta, com ara que hi ha atributs referents al mateix concepte amb noms diferents, les abreviacions de monedes, les dates i altres unitats no estan estandarditzades. Els camps no tenen cap tipus de dades associat, per tant s'ha de deduir en funció del contingut. També hi ha bastanta informació emmagatzemada en taules que es podria presentar com una plantilla. Tot hi aquests inconvenients, es pot extreure molta informació de les plantilles, realitzant abans un cert preprocessament dels seus atributs per estandarditzar el seu contingut.

2.7 PÀGINES DE DISCUSSIÓ

Les pàgines de discussió son pàgines associades a un article, on els col·laboradors a la Wikipedia exposen el seu punt de vista sobre possibles modificacions a l'article que estan associades. Normalment es proposen ampliacions, o si l'article ha sobrepassat el tema del títol original, es proposen reestructuracions o canvis de títol; o consideracions sobre la qualitat o mancances de l'article. Per il·lustrar-ho, considerem el següent extracte de la pàgina de discussió de l'article "Propulsión a chorro":

> He encontrado que algunos errores de ortografía han sido corregidos, así como algunos otros detalles. Releyendo el artículo, encuentro que Hay ideas que se repiten a lo largo del mismo de manera excesiva. El artículo descansa demasiado en la propulsión de vehiculos estelares No se concentra en torno al título que encabeza el artículo No siendo experto en temas de propulsión a chorro, me es muy difícil corregir este artículo, en particular en lo referente al tercer punto. ... Jtico (discusión) 15:20 10 ago 2007 (CEST) Título incorrecto [editar]
>
> Este artículo debe llamarse propulsión espacial y no propulsión a chorro, pues trata sobre lo primero y no sobre lo segundo. (...) 3coma14 (discusión) 12:25 12 mar 2009 (UTC)

En aquest extracte es pot veure un anàlisi sobre la qualitat gramatical i literària de l'article, i un comentari sobre la idoneïtat del títol per al contingut de l'article, ja que previsiblement el contingut de l'article s'ha anat ampliant cap a un concepte mes genèric, mantenint el títol original. També hi han pàgines de discussió en elements de l'estructura de l'enciclopèdia, com ara en plantilles o categories; i també en pàgines d'usuaris, que els usuaris usen per comunicar-se entre ells. L'ús de les pàgines de discussió ha estat limitat en l'àmbit de la mineria de dades, fins ara exclusivament per a determinar mètriques de qualitat sobre les edicions realitzades a cada article.

2.8 HISTORIES D'EDICIÓ

La pestanya que es troba més a la dreta de les que es troben al cap damunt d'una pàgina de la Wikipedia, porta a l'historia d'edicions; on es pot veure cada canvi realitzat sobre l'article al que correspon. En aquesta pagina es mostra una llista de canvis que conté per defecte els 50 últims canvis. Les pàgines d'edició son una font interessant a l'hora d'extreure informació de la Wikipedia.

Part II

LA PROPOSTA

3 ARQUITECTURA

En aquest capítol s'analitza l'arquitectura del software desenvolupat des de tres punts de vista diferents. La vista funcional correspon a l'arquitectura a nivell de la funcionalitat que porta a terme. La vista per capes correspon a l'organització del software en diferents capes per afavorir la portabilitat i canviabilitat del software per a futures versions. La vista a nivell de classes correspon a l'organització que s'ha fet de les diverses peces de software en classes, dins de les capes anteriorment mencionades.

3.1 VISTA FUNCIONAL

El software està organitzat en una cadena de mòduls, on cada mòdul usa com a dades d'entrada el resultat del mòdul anterior. Les dades d'entrada en el primer mòdul de la cadena, es un dump oficial de la wikipedia, i la sortida de l'últim son les dades extretes sobre la Wikipedia, estructurades en RDF. No es necessari que s'executin tots els mòduls cada cop que s'usa el software, però si que es necessari que quan s'executi un mòdul, s'haguin executat els mòduls anterior en la cadena amb la configuració adequada. El procés constà de les següents etapes (veure figura 3):

Una primera etapa on es transforma el format del dump oficial de la Wikipedia, a un format XML adaptat a les necessitats d'aquest projecte, i també s'estandarditza el format de certs element del text, com dates, etc. El mòdul que realitza aquest procés s'anomena WikiPrep.

La segona etapa correspon a la carrega d'aquesta informació sobre la base de dades XML, per a poder-la consultar d'una manera eficient. Aquesta etapa permet carregar de forma selectiva els articles en funció de la categoria, o de si contenen una infobox; i permet afegir articles sobre la base de dades en diferents fases, de manera que si es necessita ampliar l'àmbit dels articles continguts a la base de dades, es pot tornar a executar aquesta etapa amb una configuració diferent, i carregar nous articles. El mòdul que realitza aquest procés s'anomena Split, ja que separa l'arxiu que conté tots els articles, i n'agafa només els articles que l'usuari desitgi.

La tercera etapa correspon al procés d'aprenentatge dels patrons sobre les frases on apareixen el tipus d'entitats nombrades a extreure.

En la quarta etapa s'usen els models apresos, i es torna a accedir al contingut de la wikipedia, en el conjunt d'articles en que l'usuari desitja realitzar el procés d'extracció; s'extreuen totes les instàncies de l'atribut que detecta el software en els cossos dels articles, i emmagatzema aquesta informació, estructurada usant el model de dades RDF, sobre una base de dades. El mòdul que realitza aquest procés s'anomena WikiMiner.

L'última etapa correspon a la consulta d'aquesta informació a través d'un mòdul de query SPARQL, o un mòdul extern compatible amb l'API RedLand, o be en la serialització d'aquesta, per poder ser usada en algun altre software. El mòdul que realitza aquest procés s'anomena Query.

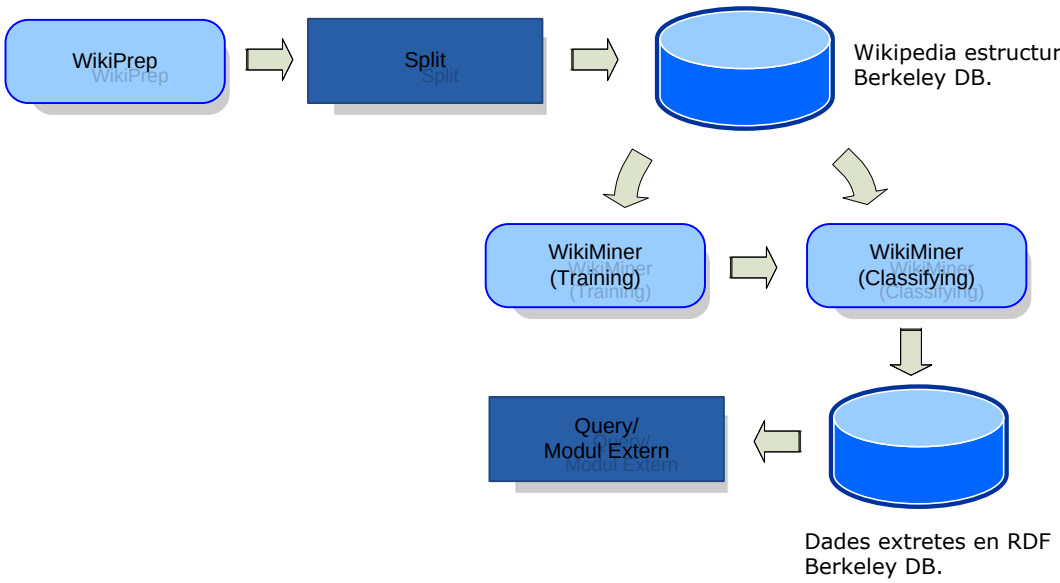


Figura 3: Vista funcional

3.2 VISTA PER CAPES

La peça principal de software d'aquest projecte, el mòdul WikiMiner, que s'encarrega de tot el procés de mineria sobre la wikipedia; està organitzat en capes. S'ha considerat el paradigma de disseny en tres capes: presentació, domini i dades. 4

La capa de dades correspon a les dos bases de dades Berkeley DB que s'usen, una per emmagatzemar el contingut de la Wikipedia, i l'altre per emmagatzemar les dades extretes de la Wikipedia. També inclou les llibreries que tradueixen els models de dades XML i RDF, sobre el model de dades de Berkeley DB, basat en taules de hash, i proporcionen una interfície per manipular, i consultar aquestes dades. També s'inclou la llibreria que facilita el tractament de les dades en format XML, tal com retorna la informació la base de dades que emmagatzema el contingut de la Wikipedia.

La capa de domini correspon, en primer lloc, a les parts del mòdul WikiMiner que s'encarreguen d'extreure dels articles seleccionats, i analitzar morfosintàcticament, usant FreeLing, les frases que son candidates potencials ja sigui pel procés d'aprenentatge com per el d'extracció d'atributs. També s'inclou en aquesta capa la lògica que implementa la funcionalitat d'aprenentatge sobre aquestes frases, per un atribut determinat. Es realitza, com s'explica en més detall en el capítol 6, una selecció d'exemples; després s'extreuen les característiques i per últim s'indueix el model que ens permetrà realitzar la classificació de tokens i el model del classificador de frases, necessària per l'extracció d'atributs. També s'inclou dins d'aquesta capa la funcionalitat relativa al procés d'extracció d'un atribut.

La capa de presentació s'encarrega de recollir els paràmetres, llegir el fitxer de configuració, i usa la capa de domini per realitzar les accions especificades per l'usuari. La capa de presentació també rep de la capa

de domini informació sobre l'estat del procés, i calcula mètriques per obtenir presicion, recall i $F - Score$.

3.3 VISTA PER CLASSES

A nivell de classes el projecte està organitzat de la següent manera: En primer lloc es disposa d'una classe que conté tota la informació necessària per gestionar un tipus de dades. Aquesta classe conté funcions per detectar tipus de dades i altres que retornen informació com ara el tipus de dades rdf que s'ha d'usar per emmagatzemar-lo. El directori data_t/Plugin conté totes les classes de tots els tipus suportats pel programa.

També hi ha una classe que gestiona la lectura del fitxer de configuració, i una altra que s'encarrega de filtrar les característiques del conjunt de vectors de característiques obtinguts, usant l'algoritme kmeans.

Per últim la lògica del programa està en dues classes que gestionen l'extracció i l'aprenentatge del classificador que indica els tokens a extreure. I una classe que filtra de cada article les frases que s'han de passar a les classes d'aprenentatge i d'extracció, i també realitza l'aprenentatge del classificador de frases, i l'usa quan s'està en mode extracció.

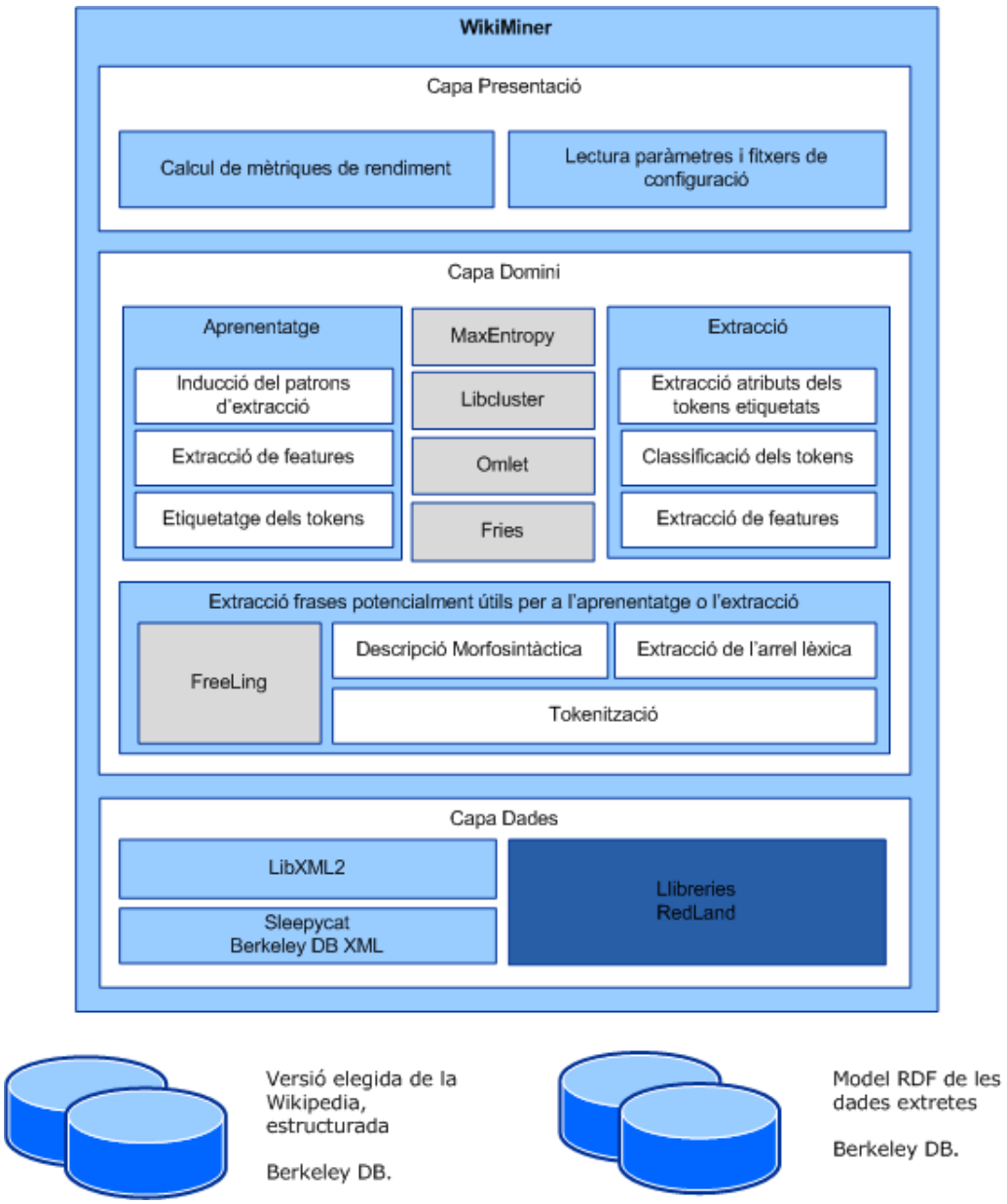


Figura 4: Arquitectura per mòduls

4 RECURSOS USATS

Per a la realització d'aquest projecte s'han usat un seguit de recursos externs al software desenvolupat, en forma de llibreries i scripts; aquest capítol es una descripció de tots els recursos que s'han usat externs al projecte.

Es necessari disposar d'accés estructurat al contingut de la Wikipedia. Degut al gran volum de dades que es poden arribar a usar com a entrada, i també degut a la mida de la majoria de versions de la Wikipedia, es necessari que el model de persistència sigui el més escalable possible. També es necessita emmagatzemar, i manipular dades en format RDF. Aquests recursos son descrits en l'apartat "Recursos per a la manipulació de dades".

El projecte fa us de tècniques per al processament del llenguatge natural, que son proporcionades a través de recursos externs. Aquests recursos estan descrits en "Recursos per al processament del llenguatge natural".

També es fa ús de recursos externs per d'obtenir accés a algoritmes d'aprenentatge automàtic supervisat, i algoritmes de clustering tal com es descriu en l'apartat "Algoritmes d'aprenentatge automàtic".

4.1 RECURSOS PER A LA MANIPULACIÓ DE DADES

4.1.1 *El paquet RedLand*

Conjunt de llibreries modulars, portables i escalables, per a la manipulació de dades estructurades usant el model RDF. També proporciona un entorn persistent i optimitzat per a la consulta, usant Berkeley DB, o un conjunt de bases de dades relacionals compatibles; amb els llenguatges SPARQL i RDQL.

L'API està escrita en C, però incorpora mòduls de nexe amb la majoria de llenguatges d'alt nivell.

Aquesta llibreria s'usa per a emmagatzemar i manipular els resultats del procés d'extracció.

4.1.2 *Sleepycat Berkeley DB*

Base de dades embedded d'alt rendiment, desenvolupada inicialment per eliminar codi propietari d'AT&T, del clon open source d'unix BSD. Actualment es mantinguda per Oracle Corporation i es distribuïda amb una llicencia dual, open source per projectes que també ho son, i de pagament pels que no ho son. El model de dades es basa en una col·lecció de parelles de valors, un amb dades a emmagatzemar, i l'altre per identificar aquestes dades; aquest model es coneix com un diccionari o taula de hash. La base de dades també permet l'accés concurrent a les dades, i l'emmagatzematge de molts tipus de dades. A més, suporta volums de dades de l'ordre dels centenars de terabytes.

Aquesta llibreria s'usa per a emmagatzemar els articles de la Wikipedia i també per a emmagatzemar els resultats del procés d'extracció.

4.1.3 *Sleepycat Berkeley DB XML*

Llibreria de middleware que permet usar Berkeley DB com una base de dades XML. Es pot considerar com una base de dades nativa en XML, ja que no ha de transformar el model XML a un model de dades diferent, com ara el model relacional; sinó que s'usa un model primitiu de dades com es el model de taules de hash, que no requereix d'una conversió complexa. Es molt similar al model de dades que s'usaria si es crees una base de dades específicament per emmagatzemar XML.

Aquesta llibreria s'usa per a emmagatzemar els articles de la Wikipedia.

4.1.4 *libxml2*

Llibreria open source sota la llicencia MIT, per al tractament de dades estructurades en XML. Originalment desenvolupada dins del projecte Gnome, es una llibreria portable i eficient, al estar implementada en C. A més, la llibreria disposa de bindings a la majoria de llenguatges d'alt nivell, i implementa interfícies estàndard en el tractament XML, com son SAX2 i DOM, amb la possibilitat d'usar XPath sobre l'estructura DOM, que proporcionen escalabilitat, en el cas del model SAX2, ideal per tractar grans volums de dades, com ara en la càrrega de la Wikipedia a la base de dades XML que es realitza en aquest projecte, i un model d'alt nivell d'abstracció que facilita el tractament de les dades, en el cas de la interfície DOM amb XPath, útil per un accés comode a les dades quan no es necessita gran capacitat d'escalabilitat, com ara en el parceig de les dades obtingudes de la base de dades.

Aquesta llibreria s'usa per a processar el resultat de la base de dades i per a carregar els articles a la Base de dades.

4.1.5 *MediaWikiDump*

Modul Perl que proporciona una interfície per a accedir al contingut d'un fitxer de dump de la wikipedia en format XML. Aquest modul permet l'accés seqüencial als articles del dump, i al contingut d'aquests articles de forma estructurada, proporcionen dades con títol, categories, etc, així com el cos de l'article.

Aquest modul s'usa en aquest projecte per preprocessar el dump de la wikipedia, i convertir-lo en el format final que usem en aquest projecte per estructurar el contingut de la wikipedia.

4.2 RECURSOS PER AL PROCESSAMENT DEL LLENGUATGE NATURAL

4.2.1 *Freeling*

Llibreria open source que proporciona serveis d'anàlisi lingüístic. Permet separar un text en tokens, on un token es una unitat lingüística bàsica, semànticament independent, que no pot unir-se amb tokens contigus sense formar una frase. Els tokens corresponen a elements estructurals del llenguatge, com ara verbs, adjectius, etc; signes de puntuació, numerals i dates, i substantius. En el cas dels substantius es reconeixen entitats nombrades de tipus organització, persona i localit-

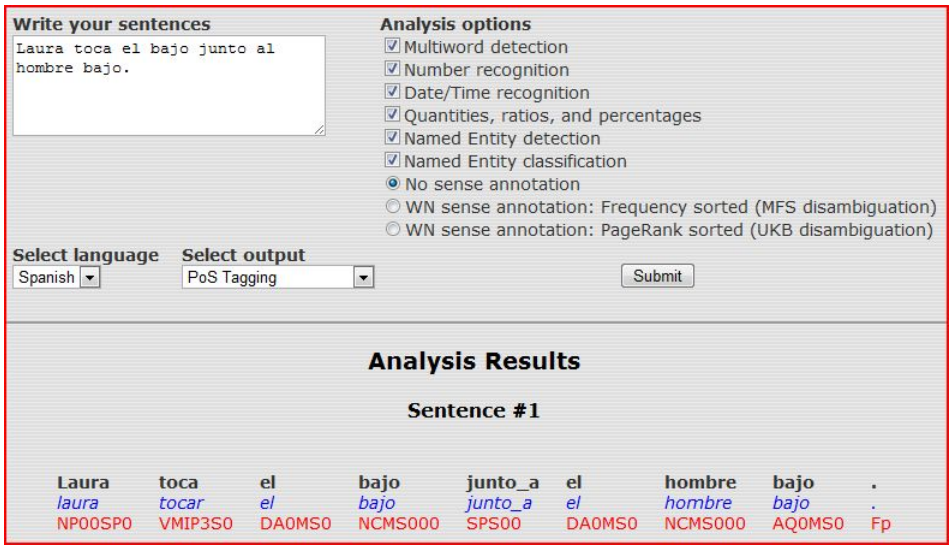


Figura 5: Aplicació de demo online de FreeLing

zació geogràfica. La llibreria també detecta les unitats (euros, metres, etc), i les dates en diferents formats.

La llibreria també determina l'arrel morfològica dels tokens, i pot generar una etiqueta de descripció morfosintàctica dels mateixos. També permet separar un text en frases, i anotar el significat de cada token en format WordNet.

Aquestes funcionalitats estan disponibles en la majoria de llenguatges suportats.

En aquest exemple es pot veure com alguns tokens consten de més d'una paraula, en el cas de "junto a", correspon a dues paraules que formen una preposició. La segona línia sota els tokens correspon als lemes; es pot observar que no en tots els casos, aquest es igual al token, per exemple en el cas de "bajo" en forma de verb, el seu lema es "bajar". La tercera línia correspon a l'etiqueta de descripció morfosintàctica de cada token; es pot apreciar com dues paraules iguals ("bajo"), tenen etiqueta diferent en funció del context, com a nom (l'instrument musical), o com a adjectiu (baix). A més es pot veure la classificació de les entitats nombrades, en el cas de la paraula "Laura" a més d'un nom detecta que es tracta d'una persona (lletres "SP" en l'etiqueta).

En el context d'aquest projecte la llibreria s'ha usat per separar el text en tokens i frases, i per generar les arrels de les paraules i les etiquetes morfosintàctiques. També s'usa per reconèixer les entitats nombrades i classificar-les en les quatre categories bàsiques.

4.3 FRIES

Llibreria open source, que forma part de FreeLing, que s'encarrega d'extreure característiques d'una cadena de dades, orientat a l'extracció de característiques de textos en llenguatge natural; utilitza el llenguatge FEX [2] i opera en cadenes de registres, on un registre son un conjunt de dades diferents sobre el mateix element (p.e. d'una paraula, el lema, l'etiqueta de descripció morfosintàctica, etc). El resultat el presenta en vectors d'enters, per a poder-lo usar com a entrada en algoritmes d'aprenentatge automàtic. Aquest recurs s'usa per a generar les característiques per al classificador de tokens.

4.4 ALGORITMES D'APRENENTATGE AUTOMÀTIC

4.4.1 *Omlet*

Llibreria open source, que forma part de FreeLing, que proporciona una interfície per a l'ús d'algoritmes d'aprenentatge automàtic, i n'incorpora un, l'AdaBoost amb Arbres de decisió. Aquest recurs s'usa per al classificador de tokens.

4.4.2 *AI::MaxEntropy*

Mòdul de perl que implementa l'algoritme de màxima entropia usant L-BFGS com a algoritme d'optimització per calcular la distribució de màxima entropia. Aquest recurs s'usa per al classificador de frases.

4.4.3 *Libcluster*

La llibreria Libcluster[6], implementa l'algoritme kmeans, i ha estat dissenyada per a analitzar mostres d'ADN. Aquesta llibreria ha estat modificada per a que retorni també les distàncies als centres dels clusters de cada vector. S'usa per a filtrar els vectors de característiques abans d'usar-los per generar el model del classificador de tokens.

4.5 ALTRES

4.5.1 *Wikiprep*

Script escrit en Perl, que processa el llenguatge Wiki d'un dump oficial de la wikipedia en format XML, i el transforma en XML, de manera que queda estructura uniformement en XML i s'elimina el markup Wiki.

Aquest script ha estat modificat per a que funcioni independentment de la versió de la wikipedia, en comptes de només en la versió anglesa, i per a que extregui les infobox dels articles, i presenti el seu contingut estructurat en XML.

4.5.2 *Swig*

Aquest programa serveix per accedir des de llenguatges interpretats, a codi escrit en C\C++, a través de la generació de wrappers.

El programa s'ha usat per accedir a FreeLing, Fries i Omlet a través de Perl. Pel que fa a Fries i Omlet, no contenien els arxius necessaris per a que Swig generés el wrapper per a Perl, per tant es van haver de crear i van sorgir un seguit de complicacions degut a la diferencia entre el funcionament intern de Perl i els executables C++, que van obligar a afegir a les llibreries de Fries i Omlet, algunes funcions que encapsulen part de l'API d'aquestes llibreries, i permeten a Swig generar els wrappers.

4.5.3 *PerlXS*

Al igual que Swig, es un llenguatge i compilador, per generar wrappers per poder usar codi C\C++, des de Perl.

S'ha usat per accedir a la llibreria Libcluster des de Perl.

5

ACCÉS ESTRUCTURAT AL CORPUS DE LA WIKIPEDIA

Aquest capítol descriu la problemàtica que suposa l'accés estructurat al corpus de la Wikipedia, i les diferents metodologies usades en aquest projecte per a realitzar aquesta tasca.

La Wikipedia es una font d'informació de dimensions considerables, semi-estructurada, i sobre virtualment qualsevol tema. Està organitzada en articles, dels quals alguns contenen Infoboxes, unes llistes de fets sobre l'article; la Wikipedia també està organitzada en categories d'articles, i conté altres fonts de meta-data com vincles entre articles, etc. Una altra característica de la Wikipedia es que es una font d'informació bastant fiable, ja que té molts usuaris que revisen el seu contingut dels quals pocs son maliciosos; a més també conté poca informació contradictòria.

Tot això fa de la Wikipedia, un recurs excel·lent tant per a la qualitat i quantitat de coneixement que se li pot extreure, com per a les facilitats que proporciona per a l'automatització del procés d'aprenentatge supervisat i per a la inducció d'una estructura ontològica.

Concretament el que ens interessarà per aquest projecte son els articles amb les seves Infoboxes, i les categories; amb l'objectiu d'automatitzar el procés d'aprenentatge supervisat.

5.1 PREPROCESSAMENT DEL DUMP DE LA WIKIPEDIA

El mètode convencional per accedir al corpus de la Wikipedia, es a través del domini wikipedia.org, que proporciona una interfície REST que permet en funció de l'URL accedir als diferents articles i categories, i també fer cerques d'articles. Aquest mètode d'accés es ideal per a l'accés a la informació per part d'humans, però no es pràctic per a l'ús del corpus des d'un sistema informàtic. Per això, també està disponible tot el corpus en els seus diferents llenguatges, per ser descarregat en diferents formats, i processat en local. Els formats disponibles son Structured Query Language (SQL) i Extensible Markup Language (XML); en el cas de la versió SQL conté tots els cossos dels articles en format wiki en una estructura de taules amb una petita part del llenguatge de marques wiki processat i emmagatzemat en atributs d'aquestes taules, com ara els vincles entre pàgines, plantilles usades en un article, etc. així com també l'estructura no continguda en el llenguatge de marques wiki, com ara si una pàgina es de re-direcció, o a quin espai de noms correspon.

Pel que fa a la versió XML, no conté res del llenguatge de marques wiki processat en XML, sinó que només consta de l'estructura no continguda en el llenguatge wiki.

Cap de les dos solucions està en un format que pugui ser usat per construir el model de base de dades que requeriria aquest projecte; per exemple, faria falta poder accedir a les infobox de cada article, i als seus atributs.

<page id="12" orglength="61516" newlength="55981" stub="0" categories="5" outlinks="333" urls="28">

<title>Elton John</title>

<categories>780754 737235 724233 771308 1920136</categories>

<links>1249918 381804 1332770 1725235 1325940 1228884 ...</links>

<urls> ... </urls>

<infoboxes>

<infobox name="Ficha de artista musical">

<attribute name="Nacimiento">[[25 de marzo]] de [[1947]] </attribute>

</infobox>

</infoboxes>

<text> Sir Elton Hercules John (n. Londres, 25 de marzo de 1947), es un cantante, compositor y pianista, británico, de música pop, rock, glam rock y piano rock, nacido como Reginald Kenneth Dwight. ...

</text> </page>

Figura 6: Extracte del format en que s'emmagatzema el contingut de la Wikipedia

Per tant, es necessari realitzar una primera etapa de preprocessament sobre qualsevol dels dos formats o processar el llenguatge wiki al accedir a l'article.

Per flexibilitat sobta per la primera opció, deixant la capa de dades sense el llenguatge d'estructuració wiki. Al final s'ha optat per usar el format XML, ja que es més senzill adaptar-lo a les necessitats d'aquest projecte, i l'autor es troba més còmode amb aquesta tecnologia. Hi ha diferents paquets de software que realitzen aquest preprocessament sobre xml; concretament s'han considerat Wiki2XML i WikiPrep.

La segona opció es més complerta, estandarditza dates i un seguit d'altres funcions que Wiki2XML no realitza, per això s'ha optat per usar WikiPrep.

Tot i ser una opció molt complerta, s'ha hagut de modificar per a que per cada article n'extregui les infobox, els seus atributs, i els valors dels atributs, i ho emmagatzemi en el xml preprocessat.

Un extracte del format final es pot veure en la figura 6.

5.2 MANIPULACIÓ DEL CORPUS DE LA WIKIPEDIA

Cada article s'emmagatzema a la base de dades en un document XML, en comptes d'un sol document per tots els articles; així es simplifica el procés d'afegir un article. Per tal de poder afegir els articles a mesura que siguin necessaris sense requerir tornar a preprocessar el dump de la Wikipedia, s'ha separat el preprocessament i la càrrega d'aquests articles a la base de dades en dos processos. El procés encarregat d'afegir articles a la base de dades s'anomena Split. Per a parcejar document

XML hi ha dues metodologies principals; una es basa en recórrer tot el document un únic cop, i anar cridant a un conjunt de funcions cada cop que comença o acaba un nou node, o es troba un bloc de text, i passar a aquestes funcions les dades necessàries per parcejar el document. Aquesta metodologia te l'avantatge que es molt ràpida i requereix poca memòria; a més escala molt bé pel que fa a la memòria en funció de la mida del document. L' altra metodologia correspon al parcejat a nivell de l'estructura DOM del document. Aquesta metodologia requereix que primer s'analitzi el document i s'emmagatzemi a memòria l'estructura d'aquest, per després poder realitzar consultes (usant XQuery o XPath) en funció de la seva estructura. Aquesta metodologia es molt més costosa tant a nivell de memòria com de velocitat, però simplifica l'extracció de dades on es requereixen realitzar consultes complexes sobre el document. La problemàtica de l'escalabilitat d'aquesta metodologia, es pot solucionar usant una base de dades, però requereix primer que es carregi el document a la pròpia base de dades per a que hi pugui treballar, per tant no es útil en tots els contextos.

En aquest projecte s'usen les dos metodologies; per preprocessar el dump de la Wikipedia, carregar el resultat a la base de dades i extreure valors concrets del resultat de consulta a la base de dades, s'usa la primera metodologia. Per altra banda, per obtenir els articles sobre els que el programa ha de treballar s'usa una base de dades, per tant la segona metodologia.

Per poder consultar el contingut de XML resultant en un interval de temps raonable, independentment a la quantitat de dades contingudes, s'ha optat per usar una base de dades XML, concretament Berkeley DB XML. S'usa el llenguatge XPath per tal de seleccionar quins articles, del total d'articles que hi han emmagatzemats a la base de dades, seran processats pel programa.

5.2.1 *Índexs*

Els índexs usats en la base de dades son:

- "name" edge-attribute-equality-string:

en l'atribut "name" de l'element "attribute" s'ha afegit un index de comparació de cadenes de text, que permet obtenir tots els atributs amb un cert nom de totes les infobox.

- "categories" node-element-substring-string:

en l'element "categories" s'ha afegit un index de comparació de part de la cadena de text. Això permet obtenir tots els articles d'una certa categoria, comparant l'element categories, amb l'id de la categoria, i després, usant una expressió regular comprovant que si avalua que conté una categoria, no sigui perquè l'id de la categoria estigui contingut en un id de categoria de valor més elevat.

```
collection("eswiki.dbxml")/page[contains(categories,$cat)
and matches(categories,"^(.*\W)?'.$cat.'(\W.*)?$")]
```

l'index permet que la funció contains sobre l'element categories sigui més escalable en funció de la càrrega; de manera que, com que la funció matches s'ha d'avaluar només en el cas que es satisfaci la funció contains, només afecta a l'escalabilitat en els casos que un id de categoria estigui contingut en un altre, que en teoria ha de ser en pocs casos.

5.2.2 *Manipulació de l'XML d'article*

Un cop s'han seleccionat els articles sobre els que treballarà el programa, s'ha d'extreure la informació que es necessita de cada article. Per tal que aquest procés sigui el més escalable possible, i tenint en compte que aquesta tasca només consisteix en l'obtenció d'uns camps, s'ha usat SAX2.

6 APRENENTATGE

Aquest capítol descriu el procés que es segueix per realitzar l'aprenentatge de les regles d'extracció d'un atribut, per que posteriorment pugui ser extret el valor d'aquest atribut.

Un cop disponible l'accés estructurat al corpus de la Wikipedia, el següent pas es generar un corpus separat en unitats lèxiques (tokens), de frases on surti l'atribut que es preténgui extreure. D'aquestes frases se n'extrauran els exemples per a l'algoritme d'aprenentatge automàtic. Per cada token també es computa el lema, i una descripció morfològica, per tal de poder generar característiques de més qualitat; Per realitzar aquest anàlisis morfològic s'usa la llibreria FreeLing[1].

Primer de tot es determina de quin tipus es l'atribut que ens intéressa, de les Infobox on apareix. No es té en compté quina Infobox es en la que apareix l'atribut, ja que les Infobox solen ser d'aspectés molt concrets (p.e. Ficha d'artista musical), i molts atributs estan compartits entre Infobox; per tant s'haurien d'aprendre molts cops si es lliguessin a una Infobox, a més a més, el volum de dades que potencialment es podria usar com a exemples per l'algoritme d'aprenentatge automàtic es reduiria significativament. El problema que té aquesta solució es que hi ha atributs que tenen el mateix nom però s'usen per conceptés diferents (p.e. Dirección, s'usa tant per indicar qui va dirigir una obra, com per indicar la direcció d'algun edifici). Ara bé els casos en que s'usa el mateix nom per dos atributs son molts pocs, per això s'ha elegit aquesta solució.

El primer pas es determinar el tipus de token que es vol extreure. Aquest projecte permet l'extracció de tokens corresponents a elements que no pertanyen al llenguatge: entitats nombrades, dates i valors numèrics, i en el cas de les entitats nombrades es separen en entitats nombrades que descriuen un indret geogràfic, una persona, una organització o cap de les anteriors; que correspon a la classificació bàsica d'entitats nombrades (la majoria de valors de les infobox son d'aquests). Per a detectar de quin tipus es l'atribut que es vol extreure, s'usen un conjunt d'objectes, un per cada tipus suportat, que avaluen el tipus d'atribut, sobre un conjunt reduït d'articles de mostra. A aquests objectes se'ls hi passa el contingut de l'atribut, per cada article de mostra, separat en tokens i analitzat morfosintàcticament, i també se'ls hi passa les frases que componen el cos de l'article. Aquestes frases s'usen per obtenir el token que s'usarà per determinar el tipus, per tant, no s'usa directament el token generat per FreeLing del valor de l'atribut, ja que aquest no disposa d'un context per poder realitzar correctament l'anàlisi morfosintàctic. Cada objecte gestiona un tipus de dades, i el primer que retorna un resultat positiu, determina el tipus de l'atribut. En aquest punt es determina si el nom de l'atribut es pot usar per avaluar la qualitat dels exemples positius, simplement buscant-lo, a l'igual que en el pas anterior, en un conjunt d'articles reduït.

S'ha usat un sistema amb tres classificadors (un per filtrar les millors frases d'on extreure l'atribut, un altre per determinar quins tokens, dels que son del mateix tipus que l'atribut, s'extreuen; i un últim que

1 http://garraf.epsevg.upc.es/freeling/

correspondria al classificador d'entitats nombrades del FreeLing), ja que el nombre d'exemples negatius de tokens son molt superiors al d'exemples positius, i per que el procés d'aprenentatge doni bons resultats, es preferible que el nombre d'exemples positius i negatius estigui equilibrat. Per això es separa el procés d'aprenentatge en dos fases, en una primera s'apren el classificador de frases, classificant-les en les que s'usaran i les que no s'usaran, per al procés d'extracció de tokens; la segona fase es l'aprenentatge del classificador de tokens, aquest procés es centra en la frase que, segons l'heurístic, conté els exemples positius per al classificador de tokens, de més qualitat.

En els següents apartats es descriu com s'obtenen els exemples positius i negatius per als classificadors de frases i de tokens.

6.1 CLASSIFICACIÓ DELS TOKENS

Es busca en els articles on apareix l'atribut en una Infobox, cadenes corresponents a un valor correcte de l'atribut, i tots els exemples trobats es consideren candidats a exemples positius i negatius. També es consideren candidats a exemples negatius els tokens que son del mateix tipus que el valor de l'atribut i que es troben al voltant del exemples considerats anteriorment.

D'aquests tokens s'extreuen les característiques usant la llibreria Fries[2], a partir de les regles d'extracció de característiques descrites en l'apartat "regles de generació de característiques". Un cop obtingudes les característiques es passen a vectors d'enters per a poder ser processades per l'algoritme d'aprenentatge automàtic.

6.1.1 *Selecció dels exemples*

Per tal de poder aplicar l'algoritme d'aprenentatge automàtic, primer es necessita un conjunt d'exemples de l'atribut que es vol extreure en el context d'una frase (exemples positius), i exemples de tokens del mateix tipus que l'atribut, però amb un valor diferent (exemples negatius).

Es poden donar casos en que un atribut contingui una llista de possibles valors; per tractar aquests casos, com que aquest programa només tracta amb atributs univaluats, un cop s'ha determinat el tipus de l'atribut, simplement consisteix en agafar el primer token del tipus de l'atribut. Només es tracten els atributs univaluats perquè per tal de determinar el valor final que se li dona a l'atribut en el procés d'extracció, s'usa la confiança donada pel classificador en tots els tokens que el classificador ha definit com a instàncies de l'atribut. També es genera el lema del token que correspon al valor de l'atribut definit a la Infobox, ja que hi ha un tipus d'atribut, les dates, que usen els lemes per comparar diferents valors.

Un cop es disposa del valor exacte de l'atribut, per un article, ja es pot passar a obtenir els exemples positius i negatius.

Primer de tot es separa, usant FreeLing, el text de l'article en tokens, es a dir, en unitats lèxiques en forma de signes de puntuació, numerals, dates, unitats gramaticals pròpies del llenguatge (verbs, preposicions, etc), o noms.

Després es segmenta el text, també a través de FreeLing, en frases, i en el cas que s'hagui determinat prèviament que el nom de l'atribut es

2 http://www.lsi.upc.edu/~nlp/omlet+fries/

útil per determinar la qualitat dels exemples, per cada frase es busca el nom de l'atribut i es seleccionen només les que el contenen. Després es generen, per cada token, els lemes i la seva etiqueta morfosintàctica. Es realitza detecció i classificació d'entitats nombrades de localitats, persones, organitzacions i de tipus variat. Per realitzar aquest anàlisis morfològic sobre el text s'usa la llibreria FreeLing.

En aquest punt, si s'ha determinat prèviament que el nom de l'atribut es útil per determinat la qualitat dels exemples, es a dir, que el nom de l'atribut es troba en el cos dels articles(p.e. el nom d'atribut "capital"es troba en el cos dels articles); s'ordenen tots els tokens que contenen el valor de l'atribut correcte, per l'article que s'està tractant, en funció de la distancia que hi hagui entre el token i el nom de l'atribut. Aquesta distància es calcula en funció del nombre de tokens del mateix tipus que l'atribut, es troben entre el nom de l'atribut i el token al qual se li està calculant la distància. A aques procés correspon al que en aquest capítol s'anomena heurístic. Per cada article s'agafa com a exemple positiu el token que té una distància menor.

Pel que fa als exemples negatius, s'agafen els tokens, del mateix tipus que l'atribut que es troben just al costat (tant dreta com esquerra) dels exemples positius. Seguint l'ordre que s'ha generat usant l'heurístic per determinar la qualitat dels exemples positius, s'agafen els exemples negatius fins que el nombre d'exemples positius i negatius es igual.

L'objectiu es mantenir equilibrat el nombre d'exemples positius i negatius, i centrar l'aprenentatge sobre els tokens que es solen trobar en les frases on es considera que hi ha els exemples de més qualitat, segons l'heurístic.

Descripció morfològica

S'usen les etiquetes EAGLES (Apèndix B) per descriure morfològicament un token. Aquestes etiquetes consisteixen en un conjunt de lletres seguides, on cada lletra representa informació morfològica del token que d'esquerra a dreta va de més abstracte a més concreta. Per exemple DAoMSo indica que es un determinant, article, el tercer caràcter es un o que no s'aplica en articles; també indica que es masculí, singular, i que no es un determinant possessiu.

6.2 CARACTERISTIQUES

Un cop estan disponibles els exemples en el context de les seves frases, i d'aquestes se n'ha extret informació a nivell morfosintàctic; el següent pas consisteix en extreure les característiques per cada exemple.

Per generar les característiques s'usa una versió enriquida del llenguatge FEX [2] implementada en la llibreria Fries.

6.2.1 *Llenguatge FEX enriquit*

FEX es un llenguatge d'extracció de característiques que opera en representacions d'observacions. Una observació conté registres, i un registre conté un conjunt variable de camps corresponent al Token, Etiqueta morfològica i Lema.

El llenguatge consta d'un conjunt de regles, que poden ser dirigides a un token concret dins d'una frase, o a tots. Per cada frase s'avaluen

REG	NOMENCLATURA	DEFINICIÓ
Paraula	w	Token
Tag	t	Descripció semàntica
Vocal	v	Actiu si el token comença per vocal
Prefix	pre	Actiu si el token conté un cert prefix
Sufix	suf	Actiu si el token conté un cert sufix
Lema	lem	Lema del token

Figura 7: Expressions RGF bàsiques

en cada token les regles actives, i les que estan actives s'inclouen en les característiques resultants.

Una regla consisteix en:

> targ [inc] [loc]: RGF [[desplaçament-esquerra, desplaçament-dret]][3]

INC: inclou el token actual.

LOC: inclou la localització relativa al token actual.

TARG: número de token dins de la frase, o -1 si es vol que s'activi per cada token.

RGF: expressió *Relation Generation Function* en forma bàsica, o com a composició d'expressions bàsiques.

DESPLAÇAMENT-ESQUERRA: mida de la finestra de tokens que s'usaran en l'expressió, cap a l'esquerra del registre actual.

DESPLAÇAMENT-DRET: mida de la finestra de tokens que s'usaran en l'expressió, cap a la dreta del registre actual.

RGFs

A la taula 7 es mostren les expressions RGF bàsiques. Existéix la possibilitat de condicionar l'activació d'un camp en funció del seu valor, si s'afegeix entre parèntesis el valor al que es vol condicionar. p.e *w(x=casa)*, només s'activarà en el cas que el token sigui *casa*.

Per exemple la regla -1 inc : t [-3,3] indica que s'extreuran els tags de tots els tokens que estiguin fins a 3 posicions cap a la dreta i cap a l'esquerra del token que s'està tractant, incloent el propi token que s'està tractant; el -1 inicial indica que la regla s'aplicarà a tots els tokens de cada frase, i no a un en particular.

Modificadors sobre les expressions RGF bàsiques

CHECKRE: Aquest modificador retorna un valor binari en funció de si l'avaluació d'una expressió regular sobre un sensor es positiva o negativa.

MATCHRE: Aquest modificador retorna el valor resultant de l'avaluació d'una expressió regular sobre un sensor.

3 Els elements entre claudators son opcionals.

CHECKMWRE: Es genera un valor resultant a partir d'un conjunt d'expressions regulars i el valor que es desitgi que retorni si el sensor encaixa en l'expressió regular.

SET: Indica si s'ha trobat un sensor en un fitxer.

SETPART: Indica si s'ha trobat part d'un sensor en un fitxer.

MAP: Transforma el valor d'un sensor en un altre en funció de les relacions especificades en un fitxer.

Per exemple la regla -1 inc : checkRE(is_acr,w,"^[A-Z][A-Z]+$") [0,0] indica que s'extreura un valor binari indicant si es compleix o no l'expressió regular aplicada sobre la forma complerta del token.

Expressions compostes

&: la característica estarà activa si almenys totes les característiques simples que formen l'expressió ho estan; i genera característiques unint registres del mateix token.

|: la característica estarà activa si almenys una de les dos, o més, característiques simples ho estan; i genera característiques unint registres del mateix token.

COLOC: genera característiques entre diversos registres actius consecutius.

SCOLOC: genera característiques entre diversos registres actius.

6.2.2 *Regles de generació de característiques*

Les regles que s'ha usat per generar les característiques que usa el classificador de tokens, es poden separar en tres grups. Un primer grup s'encarrega d'analitzar el propi token. Un altre grup analitza l'entorn més immediat al token amb informació bastant concreta, com ara la distància al token analitzat, etc. El tercer grup analitza un entorn més ampli al token a classificar, i la informació que s'extreu es més abstracte. Per a realitzar aquestes regles s'ha usat [12],[11],[10], [14] i [3].

Anàlisi del token a classificar

Aquestes característiques(descrites en la taula 1) analitzen els tokens que es volen classificar. Corresponen a un conjunt de característiques usades normalment per a aquesta tasca, i que s'usen en [3] i [12]. La primera característica s'activa si el token es un número, la segona si es un acrònim, la tercera si conté un acrònim, la cinquena si totes les lletres son majúscules i la sisena si conté un número. La setena i vuitena característica corresponen al sufix i prefix respectivament, del token; es a dir, a les tres últimes lletres i les tres primeres. La regla genera cinc possibles característiques, en funció de si el token correspon a una lletra majúscula, de més d'una lletra majúscula, d'una lletra majúscula seguida d'altres no majúscules(el token comença per majúscula), o correspon a un conjunt de dígits, o per últim, correspon a un conjunt de lletres minúscules. Les dos últimes regles generen característiques amb el lema i la forma sencera del token, respectivament.

ID	EXPRESSIÓ	FINESTRA
1	checkRE(is_num,w,"^(([0-9]+([\.,][0-9]*)?)|([0-9]{1,3}(\.[0-9]{3})*(,[0-9]*)?)|([0-9]{1,3}(,[0-9]{3})*(\.[0-9]*)?))$")	0,0
2	checkRE(is_acr,w,"^[A-Z][A-Z]+$")	0,0
3	checkRE(has_acr,w,"(([A-Z][A-Z]+_)|(_[A-Z][A-Z]+))")	0,0
4	checkRE(all_caps,w,"^([A-ZÁÉÍÓÀÈÌÒÑÇ][^(A-Z_)]*_)*[A-ZÁÉÍÓÀÈÌÒÑÇ][^(A-Z_)]*$")	0,0
5	checkRE(has_num,w,"(([0-9]+([\.,][0-9]*)?)|([0-9]{1,3}(\.[0-9]{3})*(,[0-9]*)?)|([0-9]{1,3}(,[0-9]{3})*(\.[0-9]*)?))")	0,0
6	matchRE(suf,w,"...$")	0,0
7	matchRE(pre,w,"^...")	0,0
8	checkMwRE(pat,w,"S=^[A-Z]\.?$;A=^[A-Z]+\.?$;M=^([A-ZÁÉÍÓÚÀÈÌÒÙÑÇ][a-záéíóúàèìòùñç]+\.?$);9=[0-9]+;w=^[a-záéíóúàèìòùñç]+\.?$")	0,0
9	matchRE(lem,l,"^.*$")	0,0
10	matchRE(frm,w,"^.*$")	0,0

Taula 1: Bloc de regles d'extracció de característiques que analitzen el token a classificar

Token pròxims

Aquestes característiques (descrites en la taula 2) analitzen l'entorn més immediat del token a una distància de tres tokens a dreta i esquerra. S'usa la directiva 'loc' per tal de mantenir informació sobre la posició de l'element que genera la característica. Les quatre primeres regles detecten tokens que tenen acrònims o lletres majúscules. Les tres següents generen característiques a partir dels lemes, formes complertes, i etiquetes morfosintàctiques dels tokens que es troben a l'entorn més immediat. La vuitena regla genera característiques usant una versió reduïda de l'etiqueta morfosintàctica amb informació més abstracte, a més a més, també filtra certes etiquetes, es a dir, no s'activa aquesta característica per totes les etiquetes. La novena regla genera característiques a partir de bigrames de paraules i etiquetés morfosintàctiques consecutives. La desena regla genera característiques concatenant els lemes amb les etiquetes morfosintàctiques de cada token. Aquestes característiques pertanyen al conjunt de característiques usades en el classificador d'entitats nombrades del FreeLing.

Voltants del token

Les dos primeres característiques de la taula 3, corresponen a un bag of words i bag of lemmes.

Les dos penúltimes regles generen característiques corresponents a bag of lemmes, però diferenciant entre les paraules que es troben a l'esquerra i a la dreta. Aquesta informació també està inclosa quan

ID	EXPRESSIÓ	FINESTRA
1	checkRE(ctx_isacr,w,"^[A-Z][A-Z]+$")	-3,3
2	checkRE(ctx_has_acr,w,"(([A-Z][A-Z]+_)|(_[A-Z][A-Z]+))")	-3,3
3	checkRE(ctx_allcaps,w,"^([A-ZÁÉÍÓÀÈÌÒÑÇ][^(A-Z_)]*_)*[A-ZÁÉÍÓÀÈÌÒÑÇ][^(A-Z_)]*$")	-3,3
4	checkRE(ctx_hascaps,w,"([A-ZÁÉÍÓÀÈÌÒÑÇ][^(A-Z_)]*_)*[A-ZÁÉÍÓÀÈÌÒÑÇ][^(A-Z_)]*")	-3,3
5	w	-3,3
6	l	-3,3
7	t	-3,3
8	map(st,t,"tags.dat")	-3,3
9	coloc(w,t)	-3,3
10	l&t	-3,3

Taula 2: Bloc de regles d'extracció de característiques per analitzar els tokens pròxims

ID	EXPRESSIÓ	FINESTRA
1	w	-6,6
2	l	-6,6
3	matchRE(ant_lem,l,"^.*$")	-6,0
4	matchRE(post_lem,l,"^.*$")	0,6
5	scoloc(w,t)	-6,0
6	scoloc(t,w)	0,6

Taula 3: Bloc de regles d'extracció de característiques per analitzar els voltants del token

s'usa la directiva 'loc' però també inclou informació sobre el número de paraules que hi ha entre mig. Aquestes característiques son originals per a aquest projecte.

Les dos últimes regles, generen característiques consistents en bigrames (parelles de paraules) que contenen les paraules que hi ha a l'esquerra combinades amb els tipus de paraules que hi ha entre cada paraula i el token que estem analitzant. La següent regla genera el mateix tipus de característiques però al revés, es a dir, les paraules de la dreta combinades amb els tipus de paraules que hi ha entre cada paraula i el token que s'està analitzant. Aquestes característiques son útils per detectar que hi ha cert tipus de paraula, entre el token que s'està analitzant i una paraula concreta, així es pot detectar que hi ha un token del mateix tipus que el que s'analitza abans de certa paraula, i per tant aquella paraula fa referencia a el token que està més aprop. Aquestes característiques son originals per a aquest projecte.

Gazzetters i trigger words

També s'ha inclòs un conjunt de regles que generen característiques que indiquen si s'ha trobat en els voltants del token que s'està classificant, cert tipus de paraules. Aquestes regles pertanyen al sistema de classificació d'entitats nombrades de FreeLing. Aquestes paraules corresponen a gazzetters d'indrets geogràfics, persones i organitzacions, es a dir paraules que es corresponen a indrets geogràfics, persones i organitzacions; i també trigger words també d'indrets geogràfics, persones i organitzacions, es a dir, paraules que es solen usar quan s'està parlant d'indrets geogràfics, persones i organitzacions. Aquestes regles intenten determinar el contingut semàntic del context del token a classificar.

6.3 SELECCIÓ DE LES DADES D'ENTRENAMENT

Un cop s'ha generat totes les característiques a partir dels exemples que s'han seleccionat, es filtren quines característiques de les que s'han generat i quins dels exemples dels que s'ha seleccionat , s'usaran per induir el model. El filtre sobre les característiques es realitza en funció de la distribució de probabilitats d'aquestes. Si la freqüència d'aparició d'una característica es baixa no es considera, ja que aquesta no ens serviria per induir un model generalitzat. Aquest procés es realitza a través de la llibreria fries. Per generar els vectors de característiques finals s'ha de filtrar les característiques; i per filtrar-les es necessita saber la freqüència d'aparició d'aquestes, i per tant haver generat tots els vectors de característiques. Per tant primer es generen els vectors de característiques, després la llibreria Fries calcula les freqüències i genera un diccionari amb les característiques no filtrades, un valor enter positiu que la representa, i la seva freqüència. Aquesta informació s'usa per a generar els vectors finals.

També es filtren els vectors de característiques sencers per tal d'eliminar els *outliers*, es a dir, exemples de casos que no es repeteixen, i per tant no son útils per induir un model general. Al generar els exemples de forma automàtica, aquest tipus d'exemples es poden produir, i l'adaboost es particularment sensible a aquest tipus d'exemples i tendeix a generar models que pateixen d'*overfitting* quan n'hi ha de presents.

Per tal de filtrar els exemples que generin el model més general, s'estandarditzen tots els vectors de característiques de manera que

tinguin mida igual al nombre de característiques, amb valors binaris (valor 0 si la característica no està present en el vector, i valor 1 en el cas contrari). Aquestes dades son processades per l'algoritme Kmeans, que els agrupa en conjunts en funció de la similitud entre els seus valors. A aquest algoritme també se li indica el pes que té cada posició del vector, donant més importància a les característiques que apareixen més freqüentment.

Ara bé la informació que s'usa per filtrar els vectors no es l'organització que ha generat l'algoritme kmeans, sinó un valor que usa per calcular aquesta organització, que es la distància de cada vector al centre del grup. Es filtren els vectors que tenen una distancia superior al doble de la mitja de distancies, i el resultat d'aquest procés s'usa com a entrada per AdaBoost.

6.4 CLASSIFICACIÓ DE FRASES

El classificador de frases te dos modes d'aprenentatge, en funció de si es pot usar el nom de l'atribut per determinar la qualitat d'un exemple o no, es a dir, si sol apareixer en el cos de l'article el nom de l'atribut o no. Si el nom de l'atribut es pot usar, es consideren inicialment només les frases que el contenen, i s'agafa la que conté l'exemple que s'ha catalogat com a positiu de més qualitat en el procés anterior, i s'usa com a exemple positiu, i com a exemples negatius es seleccionen els que es necessitin per balancejar el nombre de positius i negatius, donant prioritat a la selecció d'exemples que no contenen una instància de l'atribut positiu.

6.4.1 *Selecció dels exemples*

Per cada article s'agafa la frase que conté l'exemple positiu del classificador de tokens de més qualitat, i s'usa com a exemple positiu del classificador de frases. Després s'agafa un grup ordenat de frases corresponents a la frase que conté el pitjor exemple positiu per al classificador de tokens, i la resta de frases que no contenen exemples positius, i que continguin el nom de l'atribut. D'aquestes frases se n'agafen per ordre, com a exemples negatius, fins que el nombre d'exemples positius i negatius son iguals. Aquesta estratègia intenta que el classificador seleccioni les frases que contenen exemples de més qualitat. En el cas que no es pugui usar el nom de l'atribut per determinar la qualitat dels exemples, s'usen les frases amb exemples positius per al classificador de tokens, com a exemples positius per al classificador de frases, i les que contenen exclusivament exemples negatius pel classificador de tokens, com a exemples negatius, intentant que el nombre d'exemples positius i negatius sigui equilibrat.

6.4.2 *Característiques*

Les característiques usades per al classificador de frases son bàsicament un bag of lemmas, es a dir, el conjunt de lemes de les paraules que conformen la frase, sense tenir en compte la posició que ocupen en la frase. L'objectiu es eliminar informació sobre genere i número, ara bé en el cas dels verbs s'usa la forma completa, en comptes del lema per conservar el temps verbal. I en el cas de les entitats nombrades, valors

numèrics i dates, s'usa l'etiqueta morfosintàctica, per abstreure el valor. Per últim els tokens que corresponen al tipus que es vol extreure, es filtren i no es consideren com a característiques.

6.5 ALGORITMES D'APRENENTATGE AUTOMÀTIC

En aquest projecte s'usen quatre algoritmes d'aprenentatge automàtic; tres d'aprenentatge supervisat, i un no supervisat. Per a realitzar la classificació dels tokens candidats a ser extrets, s'usa el meta-algoritme AdaBoost, amb conjunció amb Arbres de decisió com a classificador simple. Per al classificador de frases s'usa el classificador de màxima entropia, i per a filtrar els vectors de característiques dels tokens s'usa kmeans.

AdaBoost

AdaBoost [4], abreviació de *Adaptative Boosting*, es un meta-algoritme d'aprenentatge que s'usa en conjunció amb altres algoritmes d'aprenentatge més senzills, per millorar el seu rendiment. Es basa en un procés iteratiu que va construint un classificador a base de diferents instàncies del classificador base.

Donada una distribució de pesos sobre el corpus d'aprenentatge, busca per cada iteració l'algoritme que dona un major rendiment (té menys errors) ponderant els exemples pel pes que tenen; i augmentant els pesos dels exemples en que ha tingut un pitjor rendiment. Al final el resultat es la suma ponderada per rendiment de cada algoritme que s'ha anat creant durant el procés iteratiu. L'algoritme aconsegueix que l'aprenentatge no es centri en els exemples més senzills, i per tant que aprengui les relacions més característiques, de manera que es redueixi considerablement la sensibilitat al *overfitting*, es a dir, que l'algoritme generalitzi i funcioni correctament fora del domini sobre el que ha estat entrenat. Tot hi això, l'algoritme es sensible al soroll i als *outliers*, es a dir, a valors que s'allunyen considerablement de la resta, per això en aquest projecte s'ha dissenyat un sistema de filtre d'exemples que intenta eliminar aquests casos, que al generar-se de forma automàtica els exemples es poden produir *outliers* en el conjunt de dades d'entrenament.

Arbres de decisió

Un arbre de decisió es un sistema de classificació que es basa en travessar un arbre *top-down* fins arribar a una fulla, on cada fulla es una categoria. Per cada node es passa sempre a un node de nivell inferior en l'estructura jeràrquica, on el procés de saltar d'un node al següent es fa en funció de les característiques de les dades d'entrada. De manera que per un conjunt de característiques, sempre s'arribaria de forma determinista al mateix node fulla.

Aquest algoritme s'usa juntament amb l'Adaboost per a classificar els tokens.

Classificador de màxima entropia

El classificador de màxima entropia es un algoritme d'aprenentatge automàtic supervisat, que es basa en un problema d'optimització condicionada, on es busca la distribució que satisfà les propietats de les característiques de les mostres, i es el més uniforme possible en els valors no especificats per les característiques, es a dir, maximitza l'entropia segons la definició de Shaon d'entropia en la informació. D'aquesta manera no s'assumeix res pels valors que no es coneixen. Suposant $x_1 ... x_n$ tot l'espai de possibles vectors de característiques per a un problema de classificació concret, llabors s'han de complir les m restriccions corresponents als vectors de característiques extrets de les mostres disponibles, per tant:

$$\sum_{i=1}^{n} Pr(x_i|I) f_k(x_i) = F_k \forall k \in (1, ..., m)$$

També s'ha d'incloure la restricció que limita la suma de les probabilitats a 1:

$$\sum_{i=1}^{n} Pr(x_i|I) = 1$$

Es podria demostrar que la distribució de Gibbs correspon a la distribució de probabilitats amb màxima entropia, i subjecte a aquest problema es pot escriure com:

$$Pr(x_i|I) = \frac{1}{\sum_{i=1}^{n} exp[\Lambda_1 f_1(x_i) + ... + \Lambda_m f_m(x_i)]} exp[\Lambda_1 f_1(x_i) + ... + \Lambda_m f_m(x_i)]$$

Per tant, per obtenir la distribució de màxima entropia s'han de determinar els paràmetres $(\Lambda_1 ... \Lambda_m)$.

Aquest algoritme s'usa en aquest projecte per al classificador de frases.

Kmeans

Kmeans [9] es un algoritme d'aprenentatge no supervisat, que agrupa un conjunt de n vectors $(x_1, ..., x_n)$, en $k < n$ conjunts $S = \{S_1, ..., S_k\}$. Es un algoritme iteratiu d'optimització, que busca la suma mínima de distancies dels vectors amb els centres dels clústers als que estan assignats; es a dir:

$$\arg\min_{S} \sum_{i=1}^{k} \sum_{x_j \in S_i} |||x_j - m_i||^2$$

On m_i es la mitja del cluster S_i.

Per cada iteració l'algoritme realitza dos passos, en un primer pas assigna cada vector al clúster el qual la distancia entre el centre del clúster i el vector es mínima:

$$S_i^{(t)} = \{x_j : ||x_j - m_i^{(t)}|| \leqslant ||x_j - m_{i'}^{(t)}|| \forall i' \in (1, ..., k)\}$$

L'últim pas calcula els centres dels nous clústers. Els valors dels centres dels clústers inicials es calculen aleatòriament o usant un heurístic.

L'algoritme necessita que se li especifiqui el nombre de clústers en que organitzar les dades.

$$m_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Aquest algoritme s'usa per obtenir un conjunt d'entrenament per al classificador de tokens, amb el minim d'*outliers*; concretament s'usen les distàncies dels vectors al centre del cluster al que han estat assignats.

7 EXTRACCIÓ

Aquest capítol descriu el procés d'extracció dels atributs, un cop s'ha realitzat l'aprenentatge dels mateixos. També descriu com s'emmagatzemen i es manipulen les dades resultants d'aquest procés d'extracció.

7.1 EXTRACCIÓ DELS ATRIBUTS

Un cop s'ha realitzar l'aprenentatge dels atributs que ens interessa extreure, el software ja està preparat per realitzar el procés d'extracció. En aquest punt s'ha de determinar de quin conjunt d'articles ens interessa extreure cada atribut, seleccionant les categories d'articles on hi pugui haver instàncies de l'atribut. El procés d'extracció segueix una estratègia similar al d'aprenentatge, es tokenitza el cos de l'article i es segmenta a nivell d'oracions. D'aquestes oracions se'n seleccionen les que contenen el nom de l'atribut, en aquest punt comencen les diferencies amb el procés d'aprenentatge; no s'etiqueten els tokens, es consideren únicament els propis tokens, i els seus respectius lemes i etiquetes morfosintàctiques com a dades d'entrada al mòdul d'extracció de característiques. El resultat del procés d'extracció de característiques s'usa com a entrada de l'algoritme de classificació Adaboost.

Si s'ha detectat que l'atribut te un nom útil per a determinar la qualitat dels exemples que s'usaran per al procés d'aprenentatge, es seleccionen només les frases que el contenen i a més, contenen algun token del mateix tipus que el valor de l'atribut. En cas que no s'hagi detectat el nom de l'atribut com a útil per determinar la qualitat dels exemples, es seleccionen totes les frases de l'article que contenen tokens del tipus que te l'atribut. Després s'usa el classificador de frases i es filtren les que retornen un resultat negatiu. En les frases amb resultat positiu, es classifiquen els tokens que corresponen al tipus de valor de l'atribut. Per tal de determinar si un token correspon al valor que s'ha d'extreure, s'agafa la confiança que retorna el classificador sobre si es tracta d'un exemple positiu, i es resta aquest valor amb la confiança que retorna sobre si es tracta d'un exemple negatiu, i s'emmagatzema cada un d'aquests valors. En el cas que la mateixa paraula doni positiu més d'un cop es suma cada un d'aquests valors. Al finalitzar aquest procés s'ordenen els valors i s'agafa la paraula amb el valor més gran. Per exemple en la frase:

> "En 1549 se fundó la primera capital de Brasil, la ciudad de Salvador, en la provincia de Bahía."

Es veu com apareix un cop la paraula "capital" que correspon al nom de l'atribut. Al costat dret hi ha el nom propi d'indret geogràfic "Brasil", que no es el valor correcte de l'atribut en l'article del qual s'ha extret aquesta oració; més a la dreta apareix "Salvador", i encara més a la dreta apareix "Bahía", cap de les dues tampoc corresponen al valor correcte de l'atribut. Per tant serien potencials exemples positius en el cas que el sistema hagués seleccionat més exemples positius que negatius. Segons l'heurístic, aquests estarien a distancia 0, 1 i 2, respectivament, ja que en el primer cas es el nom propi relatiu a indret geogràfic més próxim

a la paraula “capital"; en el segon n'hi ha un abans, i en el tercer n'hi han dos abans.

7.2 ESTRUCTURACIÓ DE LES DADES RESULTANTS

L'objectiu final d'aquest projecte es obtenir informació estructurada d'un conjunt d'articles de la Wikipedia. La informació que s'extreu de la Wikipedia consisteix en conjunts de triples <entitat, característica, valor>. Les entitats son essers, objectes o conceptes amb una sèrie de característiques associades; aquestes entitats son definides per els articles, dels quals se n'intenta extreure informació relativa a peculiaritats d'aquesta entitat, representades per valors atòmics, com ara un nombre enter en el cas de l'edat d'una persona, o be relacions amb altres entitats, com ara la capital d'un país, sent aquesta una ciutat, per tant també una entitat en sí. El model de dades que s'extreu de l'enciclopèdia, per tant, es pot formalitzar com un graf dirigit, on les arestes son atributs, i els nodes entre arestes son entitats o valors primitius.

Existeixen diferents models per representar l'estructura de dades que extreu aquest projecte; Per tal de poder consultar la informació extreta a través d'un protocol estandarditzat, s'ha optat per representar-la en RDF (Resource Description Framework)[1], un conjunt d'especificacions del World Wide Web Consortium (W3C)[2] que s'usa per al modelatge de conceptes, principalment en l'àmbit d'Internet, tot i que s'usa en tot tipus d'àmbits gracies a la seva versatilitat.

7.2.1 *Descripció de l'RDF*

RDF es un protocol per modelar conceptualment informació, a partir de meta data, es a dir, dades que descriuen la informació. El model de dades d'aquest protocol es basa en conjunts de triples consistents en un subjecte, un predicat, i un objecte. El subjecte es una entitat, el predicat denota trets o propietats del subjecte, i una relació entre el subjecte i l'objecte; i l'objecte es una altre entitat.

7.2.2 *Manipulació de l'RDF*

Per crear i consultar les dades en format RDF, s'usa la llibreria Redland que proporciona un entorn sobre una base de dades per a la manipulació d'RDF. La llibreria s'usa per emmagatzemar les dades resultants del procés d'extracció, en format RDF, per consultar-les i per serialitzar-les en un fitxer.

Aquest projecte també proporciona un entorn per a la consulta de les dades extretes usant SPARQL.

L'ús d'aquesta llibreria també permet l'ampliació d'aquest projecte amb nous mòdols de software que consultin la informació extreta, per usar-la en altres processos, que usin l'API que proporciona la llibreria.

1 http://www.w3.org/RDF/
2 http://www.w3.org/

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-
syntax-ns#" xml:base="file:resultat.rdf">
<rdf:Description rdf:about="http://upc.edu/subj/Togo">
<nso:capital xmlns:nso="http://upc.edu/obj/">Lomé</nso:capital>
</rdf:Description>
<rdf:Description rdf:about="http://upc.edu/subj/Aruba">
<nso:capital xmlns:nso="http://upc.edu/obj/">Oranjestad</nso:capital>
</rdf:Description>
<rdf:Description rdf:about="http://upc.edu/subj/Baviera»
<nso:capital xmlns:nso="http://upc.edu/obj/">Múnich</nso:capital>
</rdf:Description>
<rdf:Description rdf:about="http://upc.edu/subj/Namibia">
<nso:capital xmlns:nso="http://upc.edu/obj/">Windhoek</nso:capital>
</rdf:Description>
<rdf:Description rdf:about="http://upc.edu/subj/Toscana">
<nso:capital xmlns:nso="http://upc.edu/obj/">Florencia</nso:capital>
</rdf:Description>
</rdf:RDF>
```

Figura 8: Exemple de resultat del procés d'extracció, serialitzat en RDF

Part III

L'AVALUACIÓ

8 RENDIMENT I MÈTRIQUES D'AVALUACIÓ

Aquest capítol descriu les mètriques d'avaluació usades en aquest projecte, per tal de mesurar el rendiment que proporciona el procés d'extracció.

Per tal de mesurar el rendiment del sistema s'usen les mètriques habituals en el camp de l'extracció d'informació (Information Extraction) i la recuperació d'informació (Information Retrieval), que son: precision, recall i F-score.

Aquestes mesuren respectivament la proporció d'elements correctes respecte als extrets i la proporció d'extrets respecte els correctes; i per últim la F-score consisteix en una mitja harmònica ponderada.

Tradicionalment es posava l'èmfasi en la F-score, però actualment amb la proliferació dels motors de cerca, i el CCC (Comunity Content Creation) abans de donar informació errònia, al disposar de volums grans es prefereix prioritzar el Precision[13]. F-score permet donar més pes al recall o precision, en funció d'un paràmetre β. En aquest projecte, ens interessa donar més pes a la precisió que a la recuperació, per tant s'usa $F_{0.2}$ que dona 5 cops més pes a la precisió que al recall.

$$Precision = \frac{P_r}{P_r + P_f}$$

$$Recall = \frac{P_r}{P_r + N_f}$$

$$F_\beta = \frac{(1+\beta^2)P_r}{((1+\beta^2)P_r + \beta^2 N_f + P_f)}$$

P_r Positius reals

P_f Falsos positius

N_f Falsos negatius

β Balanceig del pes entre Precision i Recall

L'avaluació del rendiment d'una configuració de l'entorn es realitza a través del càlcul d'aquestes mètriques de manera automàtica a través del software desembolupat; també es calculen les mètriques de forma manual sobre un conjunt reduit d'exemples. No es possible una avaluació automàtica amb precisió degut al soroll inherent de les dades de la Wikipedia. Per exemple els valors de les dates de naixement en algunes ocasions son incorrectes en un any o o un dia, tant com a valor de l'atribut com a valors continguts en el cos de l'article. Això afegeix soroll

		Predicció	
		P	N
Real	P	Positiu correcte	Fals negatiu
	N	Fals positiu	Negatiu correcte

Taula 4: Predit vs real

sobre el procés d'etiquetatge, i sobre l'avaluació dels valors resultants. Ara bé l'anàlisi manual te limitacions d'escalabilitat evidents, per això es necessari sempre realitzar una avaluació de rendiment combinant el càlcul manual i l'automàtic.

8.1 CROSS-VALIDATION

Aquesta tècnica consisteix en realitzar un seguit d'iteracions on s'evalua el rendiment d'un software d'extracció o recuperació d'informació; per cada iteració s'usen dos conjunts disjunts, un per a realitzar l'aprenentatge i l'altre per a l'evaluació. Hi han diverses versions d'aquesta tècnica, la més usada s'anomena *k-fold*-on es particiona el conjunt de mostres en conjunts de la mateixa mida, i per cada iteració se n'agafa un per a l'evaluació i la resta per a l'aprenentatge, sense repetir conjunt en el procés d'evaluació entre iteracions; això permet aprofitar al màxim un conjunt de mostres limitat. Ara bé en aquest projecte el conjunt de mostres es bastant gran en la gran majoria dels casos, i usar-lo al complert suposaria invertir molt de temps quan en realitat, l'ús d'un subconjunt molt més reduït no disminuïria la precisió de l'evaluació. Per això en aquest projecte es limita a un subconjunt de 1000 articles que continguin una Infobox amb l'atribut que s'està evaluant; partit en conjunts de la mateixa mida. Un cop dividit en subconjunts, no s'usa k-fold ja que no es disposa d'un conjunt de mostres reduït, sinò que es realitzen un cert número d'avaluacions iguals al número de conjunts, on per cada avaluació s'usa un conjunt per a l'extracció, i un altre diferent per a l'aprenentatge, i no es repeteix conjunt per a l'extracció o per a l'aprenentatge entre execucions. D'aquesta manera es simplifica el query sobre la base de dades sense afectar la presició. El resultat final es la mitja de totes les avaluacions, i es un valor més fiable que realitzar una única prova.

8.2 CORBA D'APRENENTATGE

Una corba d'aprenentatge consisteix en l'avaluació del rendiment sobre conjunts de corpus usats per a l'aprenentatge de mida diferent. De manera que s'observa la variació del rendiment a mesura que augmenta la mida del conjunt d'aprenentatge. En aquesta memòria s'usa $F_{0.2}$ com a mesura de rendiment per a realitzar la corba d'aprenentatge.

9 CASOS D'ESTUDI

En aquest capítol s'estudia el comportament del software desenvolupat per a l'extracció de dates de naixement de personatges que tenen un article en la Wikipedia, i capitals de països.

Per aquestes proves no s'ha activat la funcionalitat de filtratge dels vectors de característiques, descrita en el capítol 6, ja que només millora el rendiment quan el sistema ha d'apendre un model senzill, i el conjunt d'exemples obtinguts, contenen bastant de soroll; ja que en aquestes circumstàncies pot separar clarament el soroll dels vectors que defineixen el model. Per això, si no es donen aquestes circumstàncies, aquesta funció sol empitjorar el rendiment.

9.1 DATA DE NAIXEMENT

Pel que fa a les dates de naixement, el nom de l'atribut no consta en els articles, ja que consisteix en tres paraules escrites com una de sola (fechadenacimiento), i aquest valor no apareix mai en el text. Per tant no es poden filtrar els exemples en funció de la seva qualitat. Ara bé, es una dada que apareix pocs cops en el text i a més, els cops que surt el valor, sol ser perquè s'està parlant realment de la data de naixement; per tant no disminueix el rendiment el fet de no poder realitzar un filtre en l'elecció dels exemples.

Per exemple, en el cas de l'article de Pablo Picasso, el valor de la seva data de naixement es troba en les següents dos frases:

- Pablo Ruiz Picasso (Málaga, 25 de octubre de 1881 - Mougins, Francia, 8 de marzo de 1972), conocido como Pablo Picasso, fue un pintor y escultor español; creador, junto con Georges Braque y Juan Gris, del movimiento cubista.
- Nació el 25 de octubre de 1881 en Málaga, España, en el seno de una familia pequeño burguesa.

En la primera frase, el valor es troba dins d'un parèntesis, a la dreta de la localitat de naixement; dins d'aquest parèntesi també es troba la data i localitat de defunció, separat per un guió. La segona frase conté la data de naixement també al costat de la localitat de naixement, tant país com ciutat. Es pot veure que es la data de naixement i no la de defunció, ja que conté la paraula "Nació". Aquestes frases representen una mostra del tipus d'exemples positius més habituals. Pel que fa als exemples negatius, els principals solen ser les dates de defunció, ja que es troben moltes vegades al costat de les dates de naixement, tal com es veu en el primer exemple de l'article de Picasso. També, en ocasions, al voltant de la data de naixement de la persona a la que fa referència l'article es troba la de la seva parella.

9.1.1 *Avaluació manual*

S'han detectat un nombre considerable d'articles que contenen errors en els valors de les dates de naixement, tant en el cos de l'article com

Precision	Recall	$F_{0.2} - score$
96.7	88.6	96.16

Taula 5: Rendiment de l'extracció de dates de naixement

en les Infobox. Per tant per tal d'il·lustrar el procés d'avaluació s'ha fet una avaluació manual. S'usen 200 articles per realitzar el procés d'aprenentatge, i 200 articles diferents per a l'extracció; aquests articles corresponen als 400 primers articles, que contenen l'atribut "fechadenacimiento", de la base de dades. L'orde en que s'emmagatzemen els articles correspon als identificadors dels articles, i aquests han estat assignats en funció del moment en que es va crear l'article. Per tant aquests articles no tenen cap tipus de relació entre sí, i constitueixen una mostra aleatòria d'articles amb l'atribut "fechadenacimiento". La quantitat d'articles, es va elegir intentant maximitzar el número però tenint en compte que cada article de més suposa un cost manual afegit. El rendiment de l'extracció de dates de naixement es mostra en la taula 5.

9.1.2 *Corba d'aprenentatge*

En la figura 9 es pot veure la corba d'aprenentatge de l'atribut "fechadenacimiento". La variable x correspon al nombre d'articles usats per aprendre, i la variable y correspon al rendiment, segons la mesura $F_{0.2} - score$.

Es pot observar que la corba va augmentant amb el nombre d'articles usats, fins a arribar aproximadament a 300, llabors comença a disminuir el rendiment. Per tant, es pot veure com el rendiment augmenta al augmentar el nombre d'articles, ja que el sistema va aprenent el model cada cop amb més exactitud; fins a arribar a un punt d'inflexió on canvia la dinàmica, i l'acumulació de soroll en el conjunt de dades d'entrenament precipita una disminució del rendiment al incrementar el volum de dades d'entrenament. L'algoritme de classificació de tokens es centra en els exemples més difícils de classificar, per tant, es particularment sensible al soroll.

9.2 CAPITAL D'UN PAÍS

Pel que fa a l'atribut "capital", es una entitat nombrada corresponent a un indret geogràfic. Analitzant les característiques que s'han seleccionat al realitzar l'aprenentatge sobre aquest atribut apareix la paraula çapitalï çiudad"tant com a lema com a forma complerta amb una freqüència molt elevada.

L'extracció d'aquest atribut es bastant més complexa que les dates de naixement, ja que hi han molts valors que corresponen a indrets geogràfics en un article d'un país. Per tant el nombre d'exemples negatius possibles son molt més elevats als positius, i per a que el procés d'aprenentatge del classificador de tokens funcioni correctament el nombre d'exemples positius i negatius siguin similars. A més a més, els exemples son molt més complexos. Per exemple en la majoria d'articles de països s'explica la història del país, i això inclou tots els canvis de capitals que hi han hagut. Per exemple en l'article de Sri Lanka, es troba la següent frase:

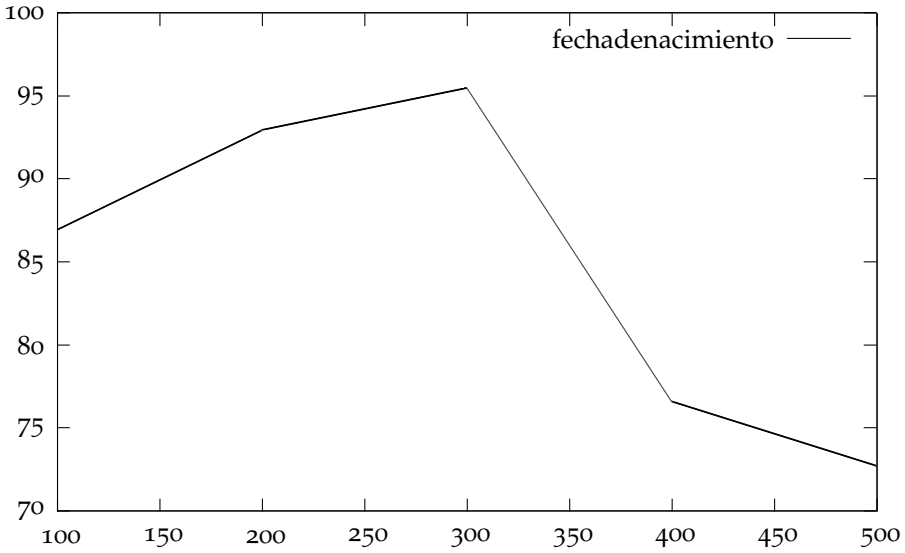


Figura 9: Corba d'aprenentatge de l'atribut fechadenacimiento

Precision	Recall	$F_{0.2} - score$
83.33	71.42	82.80

Taula 6: Rendiment de l'extracció de capitals de països sense incloure el rendiment del classificador d'entitats nombrades

> "La capital del país se mudó de Polonnaruwa a varias ciudades durante los siguientes siglos, en parte debido a las diversas invasiones extranjeras. La capital se fijó luego en Sri Jayewardenepura (Kotte) cuando las regiones costeras fueron ocupadas por los portugueses en el siglo XVI."

El valor correcte es Sri Jayewardenepura, però en la frase anterior apareix Polonnaruwa, i en tots els dos casos es parla de capital en passat. També en moltes ocasions la capital es troba al costat d'altres noms d'indrets geogràfics, que pertanyen a regions dins del país. Per tant l'extracció d'aquest atribut es una tasca més complexa que l'extracció de dates de naixement.

9.2.1 *Avaluació automàtica*

En el cas d'aquest atribut, es molt improbable que el valor de l'atribut en la Infobox, i la capital indicada en el cos de l'article, siguin valors diferents, a diferència del que passava en la data de naixement, ja que les capitals dels països son conegudes per molta gent, a diferència de les dates de naixement de personatges concrets, i per tant es difícil que un valor incorrecte es mantingui a la Wikipedia. Per això s'ha fet una avaluació automàtica, usant cross-validation amb 4 grups de 200 articles cada un. El rendiment, sense incloure el rendiment del classificador del FreeLing, es mostra en la taula 6. Ara bé, el classificador del FreeLing en tokens corresponents a indrets geogràfics en els articles de Països, presenta un Recall baix, que fa que el Recall final del sistema estigui sobre el 30

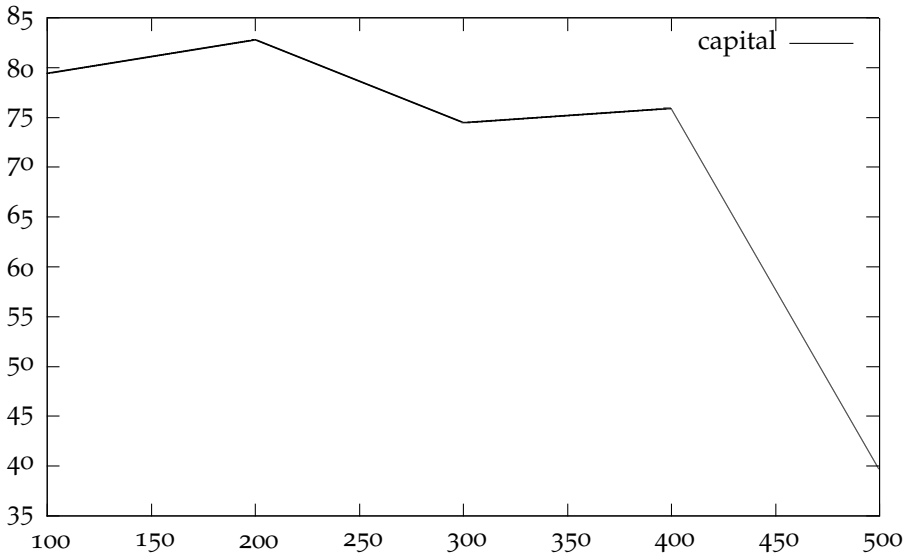


Figura 10: Corba d'aprenentatge de l'atribut capital

9.2.2 *Corba d'aprenentatge*

En la figura 10 es pot veure la corba d'aprenentatge de l'atribut "capital". La variable x correspon al nombre d'articles usats per aprendre, i la variable y correspon al rendiment, segons la mesura $F_{0.2} - score$.

L'anàlisi que es pot despendre d'aquesta corba d'aprenentatge es molt similar al fet per la corba de l'atribut "fechadenacimiento"; però en aquest cas el punt d'inflexió es troba abans, ja que la complexitat de les frases en que es troba aquest atribut es molt més elevada, i a la llarga s'incorporen al conjunt de dades d'entrenament, exemples molt complexos, que a efectes pràctics, actuen com soroll.

Part IV

CLOENDA

10 PLANIFICACIÓ I COSTOS

Aquest capítol descriu la planificació usada per portar a terme aquest projecte, i els costos teòrics de realitzar-lo.

10.1 PLANIFICACIÓ ORIGINAL

Aquest projecte tracta un tema que tot just s'està començant a investigar, i que per tant està molt poc documentant. Fer estimacions amb certesa d'un principi era una tasca complicada. Tot hi això es preveia que s'acabaria a finals d'Agost, i que la tasca principal a portar a terme era el desenvolupament de l'entorn per a d'inducció de patrons d'extracció. En la taula 8 es descriu la planificació original.

10.2 DESENVOLUPAMENT REAL

El desenvolupament del projecte (especificat en la taula 9) ha seguit la planificació en l'ordre de desenvolupament de les tàsques, exceptuant la documentació, que no s'ha realitzat al final, sinò que s'ha començat abans. Ara bé, s'ha retrassat la finalització, principalment degut a la subestimació de la complexitat del desenvolupament d'un entorn per a l'accés estructurat a la Wikipedia, i també degut a la dificultat d'estimar el temps necessari per a poder desenvolupar l'entorn d'extracció, amb un rendiment competitiu.

En la figura 11 es pot veure el diagrama de gantt, on es contrasta el desenvolupament real del projecte, amb la planificació original. Per interpretar el diagrama, s'ha de tenir en compte que cada línia correspon a una tasca, i una línia en blanc separa dos grups de tàsques. La primera línia de cada grup correspon al conjunt de tasques de tot el grup. Pel que fa als colors, el blau indica que s'ha seguit la planificació original, el verd indica que no s'ha treballat en aquella tasca, en el periode de temps que representa, ja sigui per que s'ha retrassat l'inici degut a altres tàsques, o perquè s'ha acabat abans del previst; per últim el color vermell indica que s'ha treballat en una tasca fora de l'interval de temps que tenia assignada.

10.3 COSTOS

Per el desenvolupament d'aquest projecte, es consideren dos perfils de professionals; els analistes que s'encarreguen de les parts de disseny, i els programadors, que s'encarreguen de les parts d'implementació.

Els costos de cada perfil estan descrits en la taula 7.

Perfil	EUR/h
Analista	60
Programador	30

Taula 7: Costos dels perfils

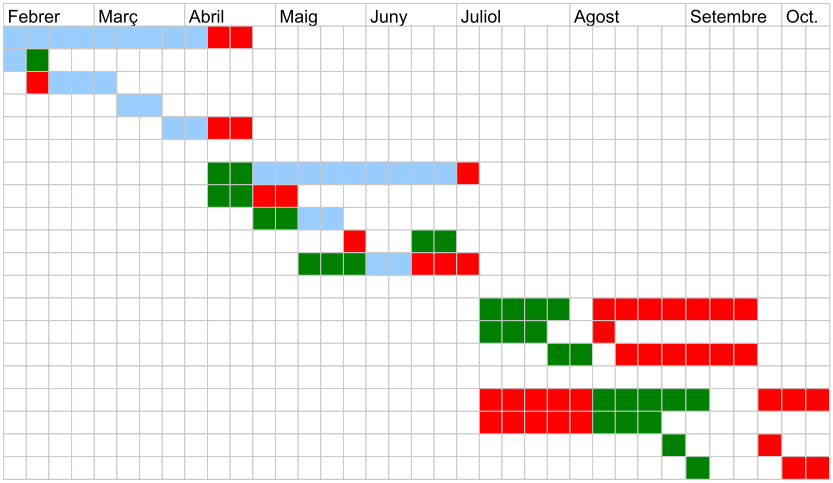


Figura 11: Gantt del desenvolupament del projecte

S'ha estimat 4 hores per dia, incloent caps de setmana.

TASCA	DURACIÓ	INICI	FINALITZACIÓ
Disseny de l'entorn per a l'accés estructurat a la Wikipedia	60	2/1/2009	4/1/2009
Anàlisis dels requisits	10	2/1/2009	2/10/2009
Anàlisis de les diverses eines disponibles	20	2/11/2009	3/2/2009
Disseny de l'entorn	10	3/3/2009	3/12/2009
Implementació de l'entorn	20	3/13/2009	4/1/2009
Disseny de l'entorn per a la mineria d'atributs bàsic	90	4/2/2009	7/4/2009
Anàlisis de requisists	20	4/2/2009	4/22/2009
Disseny de l'entorn	20	4/23/2009	6/13/2009
Implementació de l'entorn	30	4/23/2009	6/13/2009
Disseny de les regles d'extracció de característiques	20	6/14/2009	7/4/2009
Optimització de l'entorn en funció dels resultats	30	7/5/2009	8/4/2009
Modificació de l'entorn i les regles	15	7/5/2009	7/20/2009
Implementació de les modificacions	15	7/21/2009	8/4/2009
Documentació	35	8/5/2009	9/9/2009
Confecció de la memòria	20	8/5/2009	8/25/2009
Manual d'ús del software	10	8/26/2009	9/4/2009
Revisions de la memòria	5	9/5/2009	9/9/2009

Taula 8: Planificació original.

TASCA	DURACIÓ	INICI	FINALITZACIÓ
Disseny de l'entorn per a l'accés estructurat a la Wikipedia	78	2/1/2009	4/19/2009
Anàlisis dels requisits	9	2/1/2009	2/9/2009
Anàlisis de les diverses eines disponibles	23	2/10/2009	3/4/2009
Disseny de l'entorn	15	3/5/2009	3/19/2009
Implementació de l'entorn	31	3/20/2009	4/19/2009
Disseny de l'entorn per a la mineria d'atributs bàsic	71	4/20/2009	7/8/2009
Anàlisis de requisists	13	4/20/2009	5/3/2009
Disseny de l'entorn	20	5/4/2009	5/23/2009
Disseny de les regles d'extracció de característiques	6	5/24/2009	5/30/2009
Implementació de l'entorn	39	5/31/2009	7/8/2009
Optimització de l'entorn en funció dels resultats	42	8/11/2009	9/21/2009
Modificació de l'entorn i les regles	5	8/11/2009	8/15/2009
Modificació de l'entorn	37	8/16/2009	9/21/2009
Documentació	54	7/9/2009	10/12/2009
Confecció de la memòria	33	7/9/2009	8/10/2009
Manual d'ús del software	2	9/22/2009	9/23/2009
Revisions de la memòria	19	9/24/2009	10/12/2009

Taula 9: Desenvolupament real.

TASCA	HORES	EUR/H	PREU
Disseny de l'entorn per a l'accés estructurat a la Wikipedia	312		
Anàlisis dels requisits	36	60	2160
Anàlisis de les diverses eines disponibles	92	60	5520
Disseny de l'entorn	60	60	3600
Implementació de l'entorn	124	30	3720
Disseny de l'entorn per a la mineria d'atributs bàsic	284		
Anàlisis de requisists	52	60	3120
Disseny de l'entorn	80	60	4800
Disseny de les regles d'extracció de característiques	24	60	1440
Implementació de l'entorn	156	30	4680
Optimització de l'entorn en funció dels resultats	168		
Modificació de l'entorn i les regles	20	60	1200
Modificació de l'entorn	148	30	4440
Documentació	216		
Confecció de la memòria	132	30	3960
Manual d'ús del software	8	30	240
Revisions de la memòria	76	30	2280
	980		41160

Taula 10: Costos de producció

11 CONCLUSIONS

Aquest capítol presenta les conclusions extretes de la realització d'aquest projecte, i les possibilitats d'ampliació que aquest presenta.

D'aquest projecte se'n poden despendre diverses observacions. En primer lloc, es constata que la Wikipedia es un recurs molt útil per automatitzar el procés d'aprenentatge automàtic supervisat degut a la combinació de dades estructurades i no estructurades, i més concretament, degut a l'organització en articles que descriuen un únic concepte o entitat, i a les Infobox que contenen aquests, que permeten obtenir valors correctes sobre fets relatius a l'entitat sobre el que tracta l'article i per tant, poder etiquetar les instàncies d'aquest article que es troben sobre el cos de l'article automàticament. Ara bé, a la pràctica s'ha trobat que el soroll inherent a la informació pensada per ser llegida per humans que contenen les Infobox, dificulta considerablement aquesta tasca. A més, s'han de filtrar els exemples generats automàticament usant les Infobox, en funció de la seva qualitat, ja que del contrari s'afegeix complexitat innecessaria, o fins i tot, contradiccions al model, i es deteriora molt el rendiment.

Pel que fa als objectius del projecte, s'han assolit de manera satisfactòria, i tal com s'havia previst s'han hagut de realitzar successives modificacions sobre el plantejament usat per realitzar el procés d'aprenentatge i extracció, per tal d'acabar obtenint un rendiment competitiu. Per tant, en la realització d'aquest projecte s'ha dissenyat i desenvolupat un sistema capaç d'extreure peces de text del cos dels articles de la Wikipedia, corresponents al valor d'una relació no taxonòmica entre l'element extret i l'entitat definida per l'article, on aquesta relació ve definida per alguna Infobox de la Wikipedia; per tant s'ha aconseguit que el procés sigui completament automàtic, i l'usuari hagi d'especificar unicament quin es l'atribut sobre el que vol realitzar el procés d'aprenentatge i extracció, encara que si ho desitja també pot especificar altres paràmetres, com el nombre d'articles usats, etc.

Pel que fa a la part personal, he tingut la oportunitat d'aprendre molt sobre l'àrea de l'extracció d'informació, en un context molt pràctic i relativament novedos, com es la mineria sobre la Wikipedia, i més concretament l'ús de les Infobox per automatitzar el procés d'aprenentatge automàtic supervisat. També he tingut l'oportunitat d'aprendre la complexitat que entranya el disseny d'un sistema d'extracció d'informació que intenti extreure certs valors d'un text lliure, on les característiques que diferèncien aquests valors a extreure de la resta s'han d'aprendre. Tot això, i l'interés de l'autor per aquesta temàtica, han fet que la realització d'aquest projecte hagui estat una experiència molt positiva, i entretinguda.

11.1 TREBALL FUTUR

- Es podria induir una estructura ontològica a partir de la meta data que conté la Wikipedia, incloent la classificació dels articles en categories, etc. Així, es podria treballar a nivell d'Infobox, en comptes d'atribut, i s'eliminaria la restricció en l'ús d'atributs

homònims. També ampliaria el conjunt d'articles utilitzables per a l'aprenentatge, ajuntant atributs sinònims.

- Es podria extendre el projecte per a que es pogués usar també per atributs multivaluats. En aquest projecte s'ha posat aquesta restricció per poder usar la confiança donada per el classificador de tokens, per decidir en el procés d'extracció sobre un article, quin token es considera que correspon al valor correcte a extreure; i d'aquesta manera millorar considerablement el rendiment.
- Tal com fa Kylin, es podria dissenyar un mòdul independent que consultes les dades resultants del procés d'extracció emmagatzemades a la base de dades, i completes les Infobox de la Wikipedia. Això, tal com s'especifica en el [8], iniciaria una dinàmica de "bootstrapping" que permetria que futures extraccions es trobessin amb un volum de dades d'entrenament més elevat.
- Es podria adaptar a altres idiomes simplement proporcionant els arxius necessaris per a que FreeLing funcioni en l'idioma al que es vol adaptar.

Part V

APENDIX

A

MANUAL DÚS

L'ús del software resultant d'aquest projecte es realitza a través de l'execució d'un seguit de programes que retornen informació orientativa en forma de text, i es manipules a través de paràmetres proporcionats en el moment d'executar-los o a través de fitxers de configuració. El funcionament del software es tipus Batch, l'usuari no pot interactuar amb el programa un cop s'ha iniciat l'execució. Les característiques de la tasca que desenvolupa aquest software, on no es pot iniciar l'execució fins a disposar d'una configuració, i un cop iniciada l'execució no es necessita més interacció amb l'usuari; fa que aquest model en mode Batch sigui el més atractiu, ja que ademés facilita que un sistema informatic autonom, a part d'un usuari, pugui usar el software.

A.1 INSTAL.LACIÓ

Per tal de facilitar la feina als evaluadors, concient que instal·lar les dependències necessaries per executar aquest software es un procés costós, es proporciona en el DVD adjunt a aquesta memòria una imatge d'un sistema Linux bastant lleuger (Arch linux) amb totes les dependencies i el software del projecte ja instal·lat. Per tant, per tal d'usar el software del projecte hi han dues opcions que es descriuen a continuació.

A.1.1 *Ús de l'imatge*

Per tal d'usar l'imatge proporcionada al DVD, s'ha d'instal·lar el programa VirtualBox, que s'inclou en el DVD a la carpeta imatge. Es proporciona la versió per a Windows i per a Linux, si no s'usa cap d'aquests sistemes operatius a la web del programa (http://www.virtualbox.org/) es pot trobar per altres sistemes operatius. Un cop instal·lat, s'ha d'executar i elegir en el menu "File" i "Import Appliance", i indicar que la imatge a importar es troba en la carpeta imatge del DVD. Un cop importada només cal clicar a "Start" i s'executarà.

A.1.2 *Instal·lació completa*

Per tal d'instal·lar el software primer es necessari tenir instal·lades totes les llibreries que s'usen. Concretament es necessiten:

- Redland (librdf)
- Fries
- Omlet
- Freeling
- Berkeley DB
- Berkeley DB XML
- Libcluster

- AI::MaxEntropy

En el cas d'Omlet, Fries i Libcluster, es molt important usar la versió proporcionada en el DVD adjunt a aquesta memòria; ja que incorporen unes modificacions en el codi font, necessàries per executar aquest software. També s'ha de tenir en compte que al instal·lar Fries, Omlet, Freeling i Libcluster, s'ha de compilar i instal·lar per separat la llibreria de vinculació amb Perl, que es troba en la subcarpeta "perl"en el cas de Libcluster, i "API/Perl"per Fries, Omlet i Freeling.

Els moduls de Perl es poden instalar desde CPAN executant "perl -MCPAN -e install NOM"

Un cop es disposa d'un sistema amb les dependencies necessàries per executar el software, el primer que es necessita es descomprimir el tar del projecte usant la comanda tar xvzf wikiMiner.tar.gz, en el cas que ens trobem en un sistema Unix; si es un sistema Windows s'haurà d'obtenir algun software capaç de descomprimir arxius tar i gzip. Al ser axius perl es compilen quan necessites executar-los, per tant no cal compilar-los, i els arxius descomprimits, ja son el resultat de la instalació, per tant s'ha de descomprimir en algun directori no temporal.

El primer pas un cop està instal·lat es obtenir una copia del corpus de la Wikipedia sobre el que es vulgui treballar, en format xml.

A.2 OBTENCIÓ DE LA WIKIPEDIA

Per tal d'usar aquest software, es necessari primer disposar d'una copia de la versió de la wikipedia de la que es desitgui realitza el procés de mineria; accessible a través de la interficie del sistema de fitxers de la maquina en la que s'executi el software. Estan disponibles en diferents formats, copies de totes les versions de la wikipedia en diferents adreces d'internet (p.e. http://download.wikimedia.org/). Per aquest software es necessita la versió XML d'aquests arxius.

Si s'usa l'imatge proporcionada en el DVD, ja es disposa de la base de dades amb un subconjunt de la Wikipedia amb articles que contenen l'atribut çapital" i "fechadenacimiento", per tant no es necessari obtenir l'XML amb la Wikipedia.

A.3 ÚS

El projecte consta d'un seguit d'executables, pensats per ser executats en cadena on cada un usa els resultats de l'anterior.

Si s'usa l'imatge proporcionada en el DVD, es pot passar a usar directament WikiMiner, sinó, el primer pas un cop es disposa de la Wikipedia en la màquina on es vol executar el software, es executar un programa de preprocessament sobre l'arxiu XML que conté la Wikipedia. Aquest programa, tal com s'ha descrit en aquesta memòria, processa el llenguatge Wiki, i el transforma en una estructura XML adaptada a aquest projecte, per a poder ser usat en una Base de dades XML.

A.3.1 *WikiPrep*

-h: Ruta del fitxer XML amb el contingut de la Wikipedia.

A.3.2 *Split*

Un cop es disposa de la wikipedia estructurada en el format XML que usa aquest projecte, s'usa el programa Split per a carregar els articles que ens interessin a la Base de dades.

-h: Ruta del fitxer XML amb el contingut de la Wikipedia preprocessat.

-a: Només considera els articles que continguin els atributs especificats. Si se n'especifica més d'un s'han de separar per comes.

-c: Només es consideraran els articles que estiguin en alguna de les categories indicades. Si se n'especifica més d'una s'han de separar en comes.

A.3.3 *WikiMiner*

Aquest es el programa principal del projecte, i el que s'encarrega de realitzar l'aprenentatge i l'extracció dels atributs sobre la Wikipedia.

El programa accepta tres opcións d'execucuió que han d'anar sempre precedits per el nom de l'atribut a extreure:

-t: Aquest es el mode Train, s'encarrega d'induir els models per als dos classificadors usats usant el número d'articles especificat en l'arxiu de configuració, de crear el fitxer amb el lexicon que s'usa, i de crear el fitxer amb els paràmetes de l'atribut, com ara el tipus de dades que conté. etc. Per últim s'ha d'especificar sobre quin atribut es realitzarà l'aprenentatge.

La sortida estandard (STDOUT) d'aquesta opció consisteix en un missatge "TRAINING" que indica l'opció elegida, després s'especifica el nom de l'atribut i el tipus; i s'indica que es procedeix a generar el lexicon, i van apareixen els noms dels articles que processa. Un cop s'acaba de generar el lexicon, s'indica que s'està emagatzeman al disc, i que procedeix a generar els vectors de característiques. I el programa torna a mostrar cada nom d'article que processa. Un cop finalitzat indica que està filtrant els vectors, en cas que s'hagui elegit aquesta opció. Al acabar indica que està induït el model, i al finalitzar mostra un missatge "RESULTS" indicant que mostrarà un resum del procés, i a continuació indica el nombre d'exemples positius, i el de negatius. Per últim mostra el nombre d'exemples positius i negatius usats.

-x: Aquest mode s'encarrega de fer un anàlisi de rendiment usant Cross-Validation. Requereix un valor corresponent al nombre de clusters que usarà, que si no s'especifica per defecte es 1. Per últim s'ha d'especificar sobre quin atribut es faran les proves.

-c: Aquest mode s'encarrega d'extreure l'atribut especificat, usant els models induïts en execucions del programa anteriors, usant el nombre d'articles especificat en el fitxer de configuració. També hi ha l'opció -noRDF, que es pot usar en qualsevol mode d'execució que realitzi extracció, i que omet el pas d'emagatzemar el resultat en la base de dades RDF. I a més, també existeix l'opció -cats on es poden indicar les categories sobre les que es treballa; en el cas que siguin més d'una s'han de separar usant una coma. si no s'especifica aquesta opció, s'agafen els articles que contenen una Infobox amb l'atribut sobre el que es treballa.

La sortida d'aquesta opció consisteix en: un missatge indicant l'opció elegida "CLASSIFY". Va indicant el nom de l'article que està processant, els tokens que va classificant i la classe. Per últim indica el nombre

VARIABLE	DESCRIPCIÓ
ML_DIR	Directori on s'emmagatzemen els fitxers resultants del procés d'aprenentatge
N_ELEMS	Nombre d'articles que s'usaran tant per al procés d'aprenentatge com pel d'extracció
FREELING_DIR	Directori on està instal·lat el FreeLing
DB_WIKI	Directori on està allotjada la base de dades de la Wikipedia
CLUSTERING	Valor booleà que indica si s'usarà l'algoritme k-means per eliminar *outliers*

Taula 11: Variables, amb la seva descripció, del fitxer de configuració de wikiMiner

d'instàncies d'atribut extretes, el nombre d'articles sobre els que s'ha pogut extreure alguna instància, i el nombre d'articles processats.

-a: Aquest mode realitza una corba d'aprenentatge. Se li ha d'especificar el nombre d'increments que es faran sobre la mida del conjunt d'entrenament original, definida al fitxer de configuració. A continuació s'ha d'especificar el nom de l'atribut sobre el que es vol realitzar la prova. El resultat es presenta en forma de gràfica, i també es desglocen els valors de les mètriques de rendiment, per cada mida del conjunt.

A.3.4 *Query*

Aquest programa proporciona una interficie per realitzar consultes sobre les dades resultants del procés de mineria, usant el llenguatge SPARQL. El programa llegeix el contingut de l'entrada estandard (STDIN), l'interpreta com una consulta de llenguatge SPARQL i l'executa sobre la base de dades d'informació extreta i retorna el resultat per la sortida estandard (STDOUT).

A.3.5 *Serialize*

Aquest programa se'n carrega de serialitzar el contingut de la base de dades d'informació extreta, en format RDF usant XML.

-o: arxiu resultant

A.4 FITXER DE CONFIGURACIÓ DE WIKIMINER

El programa wikiMiner consta d'un fitxer de configuració per definir certs paràmetres de funcionament. Aquest fitxer es proporciona amb uns valors per defecte, que son els mateixos que assumiria el programa si no s'especifiqués la variable en aquest fitxer.

En la taula 11 es mostra els camps del fitxer de configuració, i la seva descripció. Els valors per defecte son el directori "mlData" per la variable ML_DIR, que es un directori dins del directori del programa. El directori "db" per a la variable DB_WIKI; el directori "/usr/local/share/FreeLing" per a la variable FREELING_DIR, que es el directori on s'instala per defecte FreeLing; el valor 0 per la variable CLUSTERING i el valor 200 per la variable N_ELEMS.

B

ETIQUETES EAGLES

Aquest apèndix es part de la documentació del FreeLing. Aquest conjunt d'etiquetes es basa en les etiquetes proposades pel grup EAGLES[1] per a l'anotació morfosintàctica de lexicons i corpus per a totes les llengües europees. Així, està previst que recullin els accidents gramaticals existents en les llengües europees. És per això que depenent de la llengua hi ha atributs que poden no especificar-se. Si un atribut no s'especifica significa que o bé expressa un tipus d'informació que no existeix en la llengua o que la informació no es considera rellevant. La infraespecificació d'un atribut es marca amb el 0.

A continuació es presenten, com a resum, les etiquetes que l'analitzador morfològic utilitza per al català en format de taula. Per a cada categoria es presenten els atributs, valors i codis que pot prendre, a més a més d'alguns exemples.

Les taules en les quals presentem les etiquetes de l'analitzador tenen quatre columnes. A la columna 1 trobem un número que fa referència a l'ordre i posició en què apareixen els atributs. La columna 2 fa referència als atributs, el nombre dels quals varia depenent de la categoria. A la columna 3 trobem els valors que pot prendre cada atribut i, finalment, la columna 4 representa els codis que s'han establert per a la seva representació. Les etiquetes en si mateixes només són els codis (columna 4) i se sap a quin atribut pertanyen per la posició (columna 1) en què es troben.

1 http://www.ilc.cnr.it/EAGLES96/home.html

ADJECTIUS			
Pos.	Atribut	Valor	Codi
1	Categoria	Adjectiu	A
2	Tipus	Qualificatiu	Q
		Ordinal	O
3	Grau	-	0
		Apreciatiu	A
4	Gènere	Masculí	M
		Femení	F
		Comú	C
5	Nombre	Singular	S
		Plural	P
		Invariable	N
6	Funció	-	0
		Participi	P

ADVERBIS			
Pos.	Atribut	Valor	Codi
1	Categoria	Adverbi	R
2	Tipus	General	G
		Negatiu	N

DETERMINANTS			
Pos.	Atribut	Valor	Codi
1	Categoria	Determinant	D
2	Tipus	Demostratiu	D
		Possessiu	P
		Interrogatiu	T
		Exclamatiu	E
		Indefinit	I
		Article	A
		Relatiu	R
		Numeral	N
3	Persona	Primera	1
		Segona	2
		Tercera	3
4	Gènere	Masculí	M
		Femení	F
		Comú	C
		Neutre	N
5	Nombre	Singular	S
		Plural	P
		Invariable	N
6	Posseïdor	Singular	S
		Plural	P

NOMS			
Pos.	Atribut	Valor	Codi
1	Categoria	Nom	N
2	Tipus	Comú	C
		Propi	P
3	Gènere	Masculí	M
		Femení	F
		Comú	C
4	Nombre	Singular	S
		Plural	P
		Invariable	N
5-6	Classificació semàntica	Ésser-Persona	SP
		Organització	Oo
		Lloc	Go
7	Grau	Apreciatiu	A

VERBS			
Pos.	Atribut	Valor	Codi
1	Categoria	Verb	V
2	Tipus	Principal	M
		Auxiliar	A
		Semiauxiliar	S
3	Mode	Indicatiu	I
		Subjuntiu	S
		Imperatiu	M
		Infinitiu	N
		Gerundi	G
		Participi	P
4	Temps	Present	P
		Imperfet	I
		Futur	F
		Passat	S
		Condicional	C
		-	0
5	Persona	Primera	1
		Segona	2
		Tercera	3
6	Nombre	Singular	S
		Plural	P
7	Gènere	Masculí	M
		Femení	F

PRONOMS			
Pos.	Atribut	Valor	Codi
1	Categoria	Pronom	P
2	Tipus	Personal	P
		Demostratiu	D
		Possessiu	X
		Indefinit	I
		Interrogatiu	T
		Relatiu	R
		Numeral	N
		Exclamatiu	E
3	Persona	Primera	1
		Segona	2
		Tercera	3
4	Gènere	Masculí	M
		Femení	F
		Comú	C
		Neutre	N
5	Nombre	Singular	S
		Plural	P
		Invariable	N
6	Cas	Nominatiu	N
		Acusatiu	A
		Datiu	D
		Oblic	O
7	Posseïdor	Singular	S
		Plural	P

CONJUNCIONS			
Pos.	Atribut	Valor	Codi
1	Categoria	Conjunció	C
2	Tipus	Coordinada	C
		Subordinada	S

INTERJECCIONS			
Pos.	Atribut	Valor	Codi
1	Categoria	Interjecció	I

ABREVIATURES			
Pos.	Atribut	Valor	Codi
1	Categoria	Abreviatura	Y

PREPOSICIONS			
Pos.	Atribut	Valor	Codi
1	Categoria	Adposició	S
2	Tipus	Preposició	P
3	Forma	Simple	S
		Composta	C
4	Gènere	Masculí	M
5	Nombre	Singular	S
		Plural	P

SIGNES DE PUNTUACIÓ			
Pos.	Atribut	Valor	Codi
1	Categoria	Puntuació	F

XIFRES			
Pos.	Atribut	Valor	Codi
1	Categoria	Xifra	Z
2	Tipus	Moneda	m

DATES i HORES			
Pos.	Atribut	Valor	Codi
1	Categoria	Data/Hora	W

BIBLIOGRAFIA

[1] Robert Bringhurst. *The Elements of Typographic Style*. Version 2.5. Hartley & Marks, Publishers, Point Roberts, WA, USA, 2002. (Cited on page 74.)

[2] Wen-Tau Yih. Chad Cumby. Fex user guide. version 1.2. 2003. (Cited on pages 21 and 29.)

[3] Daniel S. Weld. Fei Wu. Autonomously semantifying wikipedia. *In the Sixteenth Conference on Information and Knowledge Management*, 2007. (Cited on pages 4 and 31.)

[4] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, pages 119–139, 1997. (Cited on page 36.)

[5] Patrick W. Daly Helmut Kopka. *Guide to LaTeX*. Addison–Wesley, 4nd edition, 2003. (Cited on page 74.)

[6] Satoru Miyano Michiel de Hoon, Seiya Imoto. The c clustering library for cdna microarray data. *Laboratory of DNA Information Analysis, Human Genome Center, Institute of Medical Science, University of Tokyo*, 2009. (Cited on page 22.)

[7] David Milne Olena Medelyan, Catherine Legg and Ian H. Witten. Mining meaning from wikipedia. 2009. (Cited on pages 5 and 7.)

[8] Kayur Patel Raphael Hoffmann, Saleema Amershi. Amplifying community content creation using mixed-initiative information extraction. *In CHI*, 2009. (Cited on page 60.)

[9] II Rui Xu, Donald C. Wunsh. *Clustering*. IEEE Series on Computational Intelligence, USA, 2nd edition, 2009. (Cited on page 37.)

[10] Erik F. Tjong Kim Sang. Introduction to the conll-2002 shared task: Language-independent named entity recognition. *In: Proceedings of CoNLL-2002*, 2002. (Cited on page 31.)

[11] Lluís Marquez Xavier Carreras and Lluís Padro. Named entity extraction using adaboost. (Cited on page 31.)

[12] Lluís Padró Xavier Carreras, Lluís Màrquez. A simple named entity extractor using adaboost. *TALP Research Center*, 2002. (Cited on page 31.)

[13] Alex Xiao Li, Ye-Yi Wang. Learning query intent from regularized click graphs. *Acero Microsoft Research*, 2008. (Cited on page 45.)

[14] Antonio Toral Óscar Ferrández. Fine tuning features and post-processing rules to improve named entity recognition. *In Proceedings of the 11th International Conference on applications of Natural Language to Information Systems*, 2006. (Cited on page 31.)

COLOFÓ

Aquesta memòria ha estat confeccionada en LaTeX 2_ε [5] usant les families de fonts *Palatino* i *Euler* (fonts Type 1 PostScript *URW Palladio L* i *FPL*) de Hermann Zapf. Les llistes han estat confeccionades usant *Bera Mono*, desenvolupada originalment per Bitstream, Inc. sota el nom de"Bitstream Vera".

L'estil tipogràfic va ser inspirat per Bringhurst en *The Elements of Typographic Style* [1]. Aquest paquet d'estil està disponible per LaTeX a través de CTAN com a "`classicthesis`".

La imatge de la portada es propietat de l'usuari *ratfactor* de deviantART, distribuïda sota la llicència Creative Commons Attribution 3.0 License.