Plataforma de cerca semàntica per a hemeroteques digitals

David Quintana Carbelo

Resum—En aquest projecte es proposa una plataforma de cerca semàntica per a hemeroteques digitals basat en un Named Entity Recognition o NER. En la nostra plataforma es podrà introduir diferents textos d'articles i aquesta informació s'enviarà al nostre servidor on es processarà el text introduït per extreure els noms de persones, organitzacions, localitats i altres noms d'interès. A partir d'aquesta informació podrem veure fàcilment aquestes entitats en el text per poder així una cerca amb més profunditat a l'hemeroteca digital. En aquest projecte s'han desenvolupen cinc NER diferents i s'analitzen els seus resultats per determinar la seva eficiència respecte un arxiu de test i davant casos reals extrets a hemeroteques digitals.

Paraules clau—Named Entity Recognition, token, entitat, etiqueta, Sklearn-Crfsuite, cercador semàntic, hemeroteca digital

Abstract—This project proposes a semantic search platform for digital newspapers libraries based on a Named Entity Recognition or NER. In our platform you can enter different articles texts and this information will be sent to our server where the text entered will be processed to extract the names of people, organizations, localities and miscellaneous. From this information we can easily see these entities in the text in order to be able to search with more depth in the digital archive. Five different NER have been developed in this project and their results are analysed to determine their efficiency with a test file and with real cases extracted from digital newspaper libraries.

Index Terms—Named Entity Recognition, token, entity, label, Sklearn-Crfsuite, semantic browser, digital newspaper libraries.

1 Introducció

Actualment es coneix que les noticies que han anat apareixent en diaris i revistes es cada vegada més extensa. En un inici els diaris estaven pensats únicament per caràcter informatiu, sense tenir en compte el seu valor històric, aquests diaris va arribar a un punt que es van considerar com una part important de la historia i es va decidir conservar-los, un cop es va determinar el seu valor històric les noticies de diaris i revistes es van començar a emmagatzemar en hemeroteques, posteriorment a això es va optar a guardar totes aquestes noticies en hemeroteques digitals.

Com que cada cop hi ha més informació emmagatzemada a les hemeroteques digitals es precisa formes alternatives per poder cercar la informació que s'escau, normalment una forma de cercar informació dintre d'una hemeroteca digital és a partir d'una consulta SQL complexa, no obstant la complexitat d'aquesta consulta es pot arribar a aconseguir informació no desitjada. Una alternativa a aquestes comandes SQL seria elaborar un buscador semàntic.

En aquest treball es proposa la opció d'un buscador semàntic basat en NER, això consisteix en un algoritme que a partir d'un text introduït com a input extreu els noms de persones, organitzacions, localitats i miscel·lània.

- *E-mail de contacte: david.quintanac@e-campus.uab.cat*
- *Menció realitzada: Enginyeria de Computació.*
- *Treball tutoritzat per: Marçal Rossinyol (Centre de visió per computador)*
- *Curs 2017/18*

2 Objectius

El principal objectiu d'aquest projecte es la elaboració d'un cercador semàntic per a hemeroteques digitals. Durant l'elaboració d'aquest projecte s'han anat realitzant els següents objectius:

-O1: Estat de l'art dels NER i de les hemeroteques digitals.
  -O1.1: Aprendre diferents formes de crear un NER.
  -O1.2: Documentar-me sobre els diferents corpus per testejar un NER.
-O2: Implementar un mètode de NER.
  -O2.1: Escollir el corpus per desenvolupar el NER.
  -O2.2: Escollir l'algorisme en el qual basar el NER.
  -O2.3: Programar el NER.
-O3: Crear una interfície visual.
  -O3.1: Desenvolupar una interfície visual senzilla on es pugui enviar una frase.
  -O3.2: Comunicar la interfície amb el nostre NER.
-O4: Testejar el funcionament del NER amb textos extrets d'hemeroteques digitals.

3 ESTAT DE L'ART

Una hemeroteca consisteix en un recinte el qual s'encarrega d'emmagatzemar, conservar i classificar diaris, revistes i altres publicacions de premsa escrita.
Una hemeroteca digital segueix la mateixa funció que una hemeroteca amb l'excepció de que en comptes d'emmagatzemar aquests diaris i revistes de forma física ho fa en format digital.
Avui en dia les hemeroteques digitals van augmentant les seves bases de dades d'una manera senzilla ja que tots els diaris apareixen actualment en format digital, no obstant això els diaris més antics es poden afegir a la hemeroteca digital a partir d'escanejos de les seves pàgines i mitjançant tàctiques de OCR.
Alguns exemples d'hemeroteca digital són l'hemeroteca de La Vanguardia [1] i l'hemeroteca de El Mundo [2].
Hi ha diverses formes de consultar la informació d'aquestes hemeroteques, una forma força solida de consultar aquesta informació es a partir d'un NER.
A l'hora de elaborar un NER ens trobem dos problemes diferents, la detecció de etiquetes formades per més d'una paraula com per exemple "Banco de España" té que detectar-se amb la mateixa etiqueta com una sola entitat i l'altre problema consisteix en la classificació d'aquestes entitats, per exemple en "Banco de España" la paraula España formaria part de la organització i no s'ha de classificar com una localitat.
La avaluació del NER es determina per tres factors diferents:

- Precision: Número d'entitats que coincideixen exactament amb el conjunt d'avaluació.
- Recall: Número d'entitats que apareixen a la mateixa posició que les prediccions.
- Valor-F: Es deriva a partir de qualsevol predicció que reconegui de forma errònia un token com a part d'entitat o el classifiqui erròniament.

4 EXPLICACIÓ DELS NER UTILITZATS

En aquesta secció es farà una breu explicació del funcionament dels NER que s'han testejat en aquest projecte.
Aquests NER estan testejats amb la conll 2002 i etiquetaran el text pla amb les etiquetes que venen per defecte en la base de dades, aquestes són:

- B-PER: Indica l'inici del nom d'una persona.
- I-PER: Indica la continuació del nom d'una persona.
- B-ORG: Indica l'inici del nom d'una organització.
- I-ORG: Indica la continuació del nom d'una organització.
- B-LOC: Indica l'inici del nom d'una localitat.
- I-LOC: Indica la continuació del nom d'una localitat.
- B-MISC: Indica l'inici d'una paraula denominada com miscel·lània.
- I-MISC: Indica la continuació d'una paraula denominada com miscel·lània.
- O: Indica totes les paraules que no han d'estar etiquetades en el text.

4.1 Explicació del NER basat en diccionaris

Aquest NER només classificarà les paraules etiquetades com a entitat del training, primerament recull totes les paraules que estan taguejades i les afegeix a un diccionari, després processa el test línia a línia, assignant una etiqueta a les paraules que es troben en el diccionari.
Realitzant aquest procediment entrem en el problema de que la paraula "de" es classifica com "I-ORG" degut a que el nostre diccionari estarà etiquetada així per entitats com Banco [B-ORG] de [I-ORG] Sabadell [I-ORG].
Aquest NER esta d'exemple en la conll 2002.

4.2 Explicació del NER probabilístic

Aquest NER està basat en probabilitats, primerament s'analitzarà el nostre train aconseguint informació de quantes vegades ha aparegut un tag en el nostre train, les vegades que s'ha transicionat de un tag a un altre, la probabilitat de que una paraula pertanyi a cadascun dels tags, la probabilitat de que un tag sigui predecessor de un altre tag i la probabilitat de que un tag sigui inici d'una frase.
Un cop tenim això extraiem la possibilitat més alta de que una paraula tingui un tag determinat i la classifiquem, aquesta classificació té en compte les paraules d'abans de la paraula etiquetada i el tag de la paraula anterior.
Les paraules que nomes apareixen una vegada en el nostre train es tenen en compte com "unknown" per així tenir una referencia per classificar paraules que apareixen en el test i no en el train.

4.3 Explicació NER seqüencial

S'han implementat dos NER seqüencials cridant a les funcions que etiqueten el text pla.

4.3.1 Explicació Polyglot

El sistema està basat en polyglot, aquest mètode aprèn representacions de paraules distribuïdes (mots incrustats) que codifiques les característiques semàntiques i sintàctiques de les paraules en cada idioma.
El sistema genera automàticament conjunts de dades de l'estructura d'enllaç de Wikipedia i es atributs de Freebase.
El sistema aplica dues etapes de processament (sobreexplotació i concordança exacta de superfície) que no requereixen coneixements lingüístics.
Aquest sistema està implementat en el següent lloc web [3].
Aquest NER s'ha desenvolupat cridant a les funcions que analitzen el text pla.

4.3.2 Explicació Sklearn-Crfsuite

Aquest NER utilitza les llibreries de sklearn-crfsuite, es pot implementar fàcilment seguint el següent tutorial [4], podem veure una documentació mes detallada en el següent enllaç [5].
Sklearn-Crfsuite es basa en un sistema de cross-validation [6] i hyperparameter optimization.
Aquest NER s'ha desenvolupat cridant a les funcions que analitzen el text pla.

5 FORMALITZACIÓ DEL NER PROBABILÍSTIC

A continuació s'exposarà una formalització del NER probabilístic.

Primerament extraurem la quantitat de cada etiqueta que apareix en el nostre train, per realitzar aquesta tasca generarem un hasheable on extraurem cada paraula del nostre train X amb la seva etiqueta assignada Y, si X i Y no existeixen en el hasheable sels designarà un 1 com la seva quantitat Z, sinó Z s'incrementarà en 1. Un cop fet això s'introduirà la paraula UNK amb cadascuna de les etiquetes Y on la seva quantitat Z serà la suma de tots els inputs del nostre hasheable on Z=1. Finalment s'introduirà en un array anomenat NER-arr la quantitat de cada una de les etiquetes del nostre train.

A continuació s'extraurà la quantitat d'etiquetes de transició del nostre train, aquesta informació es guardarà en un array anomenat tranNER.

El següent que extraurem serà lexical_prob que consisteix en un diccionari on apareixerà cada paraula del nostre lexicon amb les seves etiquetes i la probabilitat de tenir aquesta etiqueta.

El següent que extraurem serà transitional_prob que consisteix en una estructura igual que lexical_prob però en comptes de ferla a partir del nostre lexicon es farà a partir de les nostres paraules en transició.

El següent que es farà es aplicar smoothing a la variable transitional_prob.

Per tenir en compte les paraules que apareixen a l'inici de cada frase es crearà NER_start_prob que consisteix en una estructura com la de lexical_prob però amb les paraules d'inici de frase.

Finalment s'analitzarà el nostre test a partir de la següent formula:

```
Score = (transitional_prob[prevNER][ner] * lexical_prob[word][ner])
```

Designant cada etiqueta al valor amb score més alt.

6 INTERFÍCIE DE LA DEMO

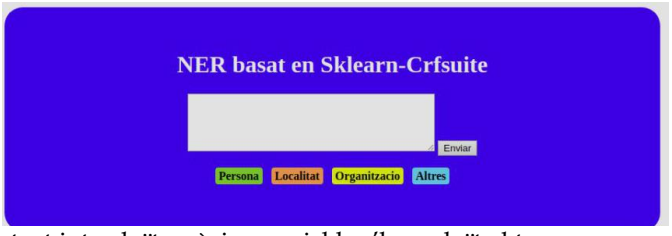
Per testejar el nostre NER s'ha creat una interfície web amb un servidor Xampp.

Quan el client web envia una frase a analitzar crea un socket on mana la informació a localholst pel port 10.000, aquesta frase es manarà codificada en bits.

Per rebre aquesta frase estarà el nostre servidor executant-se, el servidor esta escoltant de fora continua totes les peticions que vinguin a per localhost pel port 10.000 a través de sockets. El servidor es capaç de rebre 5 peticions simultànies fent que el NER extregui les entitats del text.

S'han creat dues demos per extreure resultats del text, la primera consisteix en el NER basat en probabilitats, que al estar programat de zero es la que es va elaborar en un inici. La segona demo es va crear a partir de veure l'eficàcia que tenia el NER basat en Sklearn-Crfsuite, introduint el NER basat en Sklearn-Crfsuite sense postag en un servidor basat en sockets com el de la primera demo.

Per tractar el text introduït en la interfície web la segona demo primerament separa les paraules utilitzant toktok-Tokenizer, aquesta llibreria esta basada en textos en castellà per lo que tracta correctament les paraules accentuades durant la codificació i els símbols "¿" i "¡" a mes de la lletra "ñ". Per estalviar temps d'execució a l'hora de tractar el text introduït s'ha guardat la informació necessària per etiquetar el nostre text mitjançant "cpickle" que consisteix en una eina que guardarà la variable en un arxiu .p que es carregarà en una variable quan anem a tractar el



text introduït, gràcies a cpickle s'ha reduït el temps que es trigava en processar la variable de train en aproximadament un 90%.

Un cop tractada la frase es retornarà al client on es mostrarà amb les paraules etiquetades segons el criteri del NER, la resposta que retorna el servidor ha de ser d'un màxim de 50.000 paraules.

L'aspecte de la demo es el de la imatge 1.

Imatge 1

7 RESULTATS I COMPARATIVES ENTRE ELS NER

A continuació s'analitzaran els resultats obtinguts amb els diferents NER implementats respecte el fitxer de test de la Conll 2002.

En la següent taula es poden apreciar els diferents resultats amb els diferents NER testejats:

	Probabilístic	Diccionari	Polyglot	Sklearn-crfsuite	Sklearn-crfsuite sense postag
LOC	45%	64.04%	66.23%	79.10%	78.20%
MISC	5.17%	32.48%	0%	57.38%	55.59%
ORG	32.04%	26.23%	41.76%	78.79%	77.78%
PER	45.78%	22.51%	74.58%	86.13%	84.32%
Average	25.46%	34.73%	56.29%	78.62%	77.44%

- El NER que esta obtenint millors resultats es el Sklearn-Crfsuite, no obstant això el temps d'execució es una mica superior als altres NER utilitzats en aquest projecte. No obstant això podem deduir que els sistemes de cross-validation amb hyperparameter optimization dona uns resultats molt òptims a l'hora de desenvolupar un NER.
- El NER basat en Sklearn-Crfsuite sense postagdona resultats molt similars al de Sklearn-Crfsuite. Veient això podem veure que no es un problema el desenvolupar un NER amb Sklearn-Crfsuite en conjunt de dades sense postag.
- El NER basat en Polyglot dona uns bons resultats, el seu average es menor que el de Sklearn-Crfsuite, però hem de tenir en compte que Polyglot no detecta la miscel·lània i al tenir un zero en aquest camp el seu average baixa considerablement.
- Tant el NER basat en probabilitats com el NER basat en diccionaris donen uns pitjors resultats que Polyglot o Sklearn-Crfsuite, no obstant això aquest dos NER són nomes n fitxer de python mentre que Sklearn-Crfsuite i Polyglot consisteixen en un entramat

de fitxers mes extens.
- El NER basat en diccionaris dona uns resultats similars al NER probabilístic, no obstant això el NER basat en diccionaris classificarà d'una pitjor forma un text que contingui paraules no incloses en el train.
- El NER probabilístic classifica millor les persones i les organitzacions que el NER basat en diccionaris, això es degut a que no només mira la paraula en si sinó les probabilitats que tingui un tag determinat.
- El NER basat en diccionaris detecta millor els "MISC" que el NER probabilístic, això es degut a la possible ambigüitat d'aquest tag, on la classificació de "MISC" avarca una gran quantitat d'opcions.
- El NER basat en diccionaris detecta millor les localitats que el probabilístic, això es degut a que al aparèixer en el test directament el clàssica com a localitat mentre que el probabilístic el pot arribar a classificar amb un altre tag depenent de la probabilitat obtinguda.

8 Resultats amb titulars d'articles

A continuació s'analitzarà el resultat que s'ha obtingut amb els diferents NER amb titulars de diaris.
En l'extracció de les entitats s'ha marcat de la següent forma cada una de les etiquetes:

- Localitat: [B-LOC], [I-LOC]
- Persona: [B-PER], [I-PER]
- Organització: [B-ORG], [I-ORG]
- Miscel·lània: [B-MISC], [I-MISC]

Frase 1:

Probabilístic:
La fuga del penal de San [B-LOC] Cristòbal, una historia de la Guerra [B-MISC] Civil Española.

Diccionaris:
La fuga del penal de [B-ORG] San Cristòbal, una historia de [B-ORG] la Guerra [B-MISC] Civil [I-MISC] Española.

Polyglot:
La fuga del penal de San [B-LOC] Cristòbal [I-LOC], una historia de la Guerra Civil Española.

Sklearn-Crfsuite:
La fuga del penal de San [B-LOC] Cristòbal [I-LOC], una historia de la Guerra [B-MISC] Civil [I-MISC] Española [I-MISC].

Sklearn-Crfsuite sense postag:
La fuga del penal de San [B-LOC] Cristòbal [I-LOC], una historia de la Guerra [B-MISC] Civil [I-MISC] Española [I-MISC].

Frase 2:

Probabilístic:
La moción de Sánchez dispara la prima de riesgo y tumba el IBEX [B-MISC].

Diccionaris:
La moción de [B-ORG] Sánchez dispara la prima de [B-ORG] riesgo y tumba el IBEX.

Polyglot:
La moción de Sánchez [B-PER] dispara la prima de riesgo y tumba el IBEX.

Sklearn-Crfsuite:
La moción de Sánchez [B-PER] dispara la prima de riesgo y tumba el IBEX [B-MISC].

Sklearn-Crfsuite sense postag:
La moción de Sánchez [B-PER] dispara la prima de riesgo y tumba el IBEX [B-MISC].

Frase 3:

Probabilístic:
El PSOE [B-ORG] registra en el Congreso [B-ORG] una moción de censura contra Mariano [B-PER] Rajoy [I-PER].

Diccionaris:
El PSOE [B-ORG] registra en el Congreso [B-ORG] una moción de [B-ORG] censura contra Mariano [B-PER] Rajoy [I-PER].

Polyglot:
El PSOE [B-ORG] registra en el Congreso una moción de censura contra Mariano [B-PER] Rajoy [I-PER].

Sklearn-Crfsuite:
El PSOE [B-ORG] registra en el Congreso una moción de censura contra Mariano [B-PER] Rajoy [I-PER].

Sklearn-Crfsuite sense postag:
El PSOE [B-ORG] registra en el Congreso una moción de censura contra Mariano [B-PER] Rajoy [I-PER].

Frase 4:

Probabilístic:
La Bolsa [B-ORG], rezagada en más de 3 puntos frente a Europa [B-LOC].

Diccionaris:
La Bolsa, rezagada en más de [B-ORG] 3 puntos frente a Europa [B-LOC].

Polyglot:
La Bolsa, rezagada en más de 3 puntos frente a Europa [B-LOC].

Sklearn-Crfsuite:
La Bolsa [B-ORG], rezagada en más de 3 puntos frente a

Europa [B-LOC].

Sklearn-Crfsuite sense postag:
La Bolsa [B-ORG], rezagada en más de 3 puntos frente a Europa [B-LOC].

Frase 5:

Probabilístic:
Harvey Weinstein quedará en libertad vigilada con una fianza de un millón de dolares.

Diccionaris:
Harvey Weinstein quedará en libertad vigilada con una fianza de [B-ORG] un millón de [B-ORG] dolares.

Polyglot:
Harvey [B-PER] Weinstein [I-PER] quedará en libertad vigilada con una fianza de un millón de dolares.

Probabilístic:
Harvey [B-PER] Weinstein [I-PER] quedará en libertad vigilada con una fianza de un millón de dolares.

Probabilístic:
Harvey [B-PER] Weinstein [I-PER] quedará en libertad vigilada con una fianza de un millón de dolares.

Ànalisi dels titulars:

- En casos reals podem observar com el NER basat en probabilitats detecta gran quantitat de les entitats que apareixen, en la frase 1 podem veure que les entitats que estan composades per diverses paraules a vegades no les detecta correctament, per solucionar això es podria "augmentar" de forma controlada el score en els tags a l'hora de detectar a quin tag correspon, exceptuant el tag "O".
- El NER basat en diccionaris detecta totes les paraules "de" com a organització, aquest NER únicament extreu tots els tags del train i despès mira cada una de les paraules del test per si estaven etiquetades en el train.
- El Polyglot dona uns bons resultats, exceptuant la mancança de que no detecta la miscel·lània, ja que, no està implementada.
- El NER basat en Sklearn-Crfsuite i SklearnCrfsuite sense postag son els que donen uns millors resultats, a la frase 3 no detecta la paraula "Congreso" com a una organització. En aquesta frase aquesta paraula podria ser miscel·lània en comptes d'organització.
- El NER basat en Sklearn-Crfsuite i Sklearn-Crfsuite sense postag donen els mateixos resultats, aquest comportament es podia arribar a deduir a partir dels resultats mostrats en l'apartat anterior d'aquest informe.

- El NER probabilístic té dificultats per detectar noms o

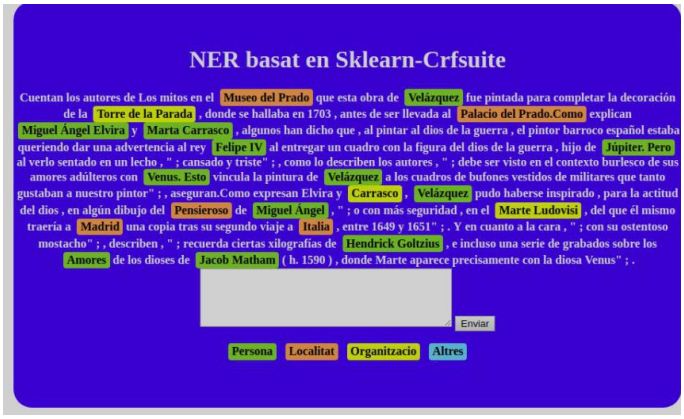
localitats desconegudes en el seu train ja que les tracta com a unknown, aquesta deducció es pot apreciar a la frase 5.

- El NER basat en diccionaris té el greu problema de marcar paraules que no han d'estar etiquetades com a entitats, mentre que els altres quatre NER rarament marquen alguna entitat on no hi ha res a marcar.

9 Resultats amb el cos d'un article

Aquests son alguns resultats aconseguits amb el cos d'un article:

Imatge 2
Imatge 3



Imatge 4
Imatge 5

Anàlisi dels textos:

- Com podem observar el processament en textos te força errades a l'hora de classificar aquests, un problema que es força freqüent es la concatenació d'un tag on es continua etiquetant part del text que no pertany al tag. La gran majoria de vegades això succeeix quan l'etiqueta acaba en punt.
- Un altre error que esta realitzant es l'incorrecte etiquetació de les entitats, sobretot en paraules que no apareixen en el training i que pel context l'algorisme dubta a quin dels labels pertany aquesta entitat.
- Després d'un punt i coma quan la paraula comença per majúscula l'algorisme etiqueta aquest com una entitat, això es degut a que en el train no es prepara correctament al algorisme per avortar aquest tipus de situació.
- A la imatge 3 marca com a persona "Rockefeller Center.Fueron" això es degut a que al introduir en el quadre de text el text a analitzar faltava introduir un espai per lo que provoca que s'inclogui la paraula "Fueron" en el mateix tag.
- El text que no conté cap etiqueta a prop i no es cap entitat no el marca incorrectament, ja que l'algorisme te en compte si la paraula es majúscula.
- El temps de processament del text es molt ràpid, d'entre 1 i 3 segons.

Conclusions

Aquest treball m'ha servit per entendre instruir-me en el funcionament dels NER a més de seguir un treball més extens que la resta dels realitzats al llarg de la carrera.
Els objectius del treball s'han assolit correctament, no obstant això al inici del projecte vaig tenir problemes a l'hora de crear el primer NER.
La quantitat d'hores que es van calcular al inici del treball final no es van complir perquè força treball realitzat no era de la suficient qualitat i es va deixar enrere per lo que es va tenir que dedicar més temps del calculat per realitzar el projecte.
Un dels problemes més destacables a l'hora de realitzar el projecte va ser els problemes que es van tenir al utilitzar Windows, sobretot a l'hora d'instal·lar llibreries. Per abolir aquest problema es va optar a realitzar aquest treball en Ubuntu. A part de tenir menys dificultats per instal·lar els paquets necessaris per aquest projecte tenim l'avantatge de que Ubuntu és un sistema operatiu gratuït per lo que per instal·lar aquest buscador en un servidor tindria un cost més reduït.
Un altre problema va ser que python 2.7 no conté llibreries necessàries per la demo 2, el codi de Sklearn-Crfsuite s'executa en python 3.5 per lo que es va tenir que adaptar la part del codi del servidor referent als sockets per que funciones amb python 3.5, un dels canvis més rellevants d'això es que els missatges s'envien com cadenes de bits en comptes de strings i aquests s'han de descodificar per executar el nostre NER i després codificar-los de nou per retornar-los al client.
Respecte al treball futur del projecte s'hauria de solucionar els errors de les etiquetacions incorrectes i el problema de la "sobreetiquetació" en paraules que no pertanyen a l'entitat. Un altre treball per al futur seria realitzar un sistema de base de dades robust amb contingut de les hemeroteques digitals i instaurar el codi de la demo en una interfície web més elaborada per testejar el codi en la cerca en una base de dades d'una hemeroteca digital.

Agraïments

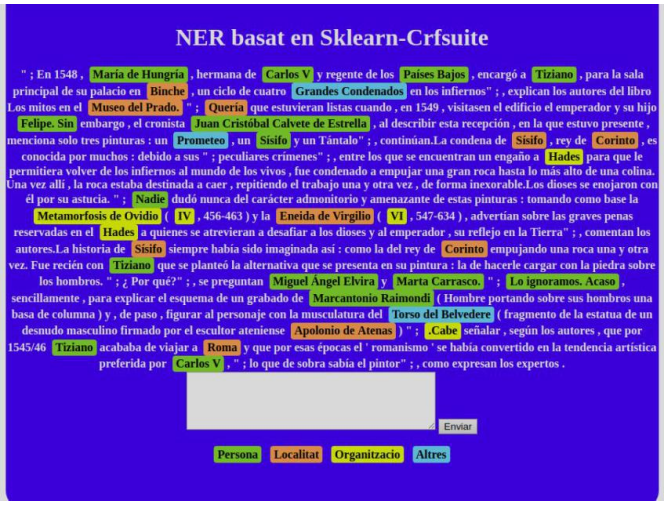
Vull agrair al meu tutor del treball final, Marçal Rossinyol, el temps que ha dedicat en el seguiment d'aquest projecte, ja que hem tingut una reunió cada setmana per revisar el progrés del treball. A més que ha resolt el dubtes que anava tenint al llarg del projecte d'una forma ràpida i eficaç.

Bibliografia

[1] http://www.lavanguardia.com/hemeroteca

[2] http://www.elmundo.es/hemeroteca/

[3]https://sites.google.com/site/rmyeid/projects/polylg



ot-ner

[4]https://sklearn-crfsuite.readthedocs.io/en/latest/tutorial.html

[5]https://media.readthedocs.org/pdf/sklearn-crfsuite/latest/sklearn-crfsuite.pdf

[6]http://scikit-learn.org/stable/modules/cross_validation.html

[7] Joseph Turian, Lev Ratinov, Yoshua Bengio (2010) Word representations: A simple and general method for semi-supervised learning,

[8] Tomas Mikolov, Wen-tau Yih, Geoffrey Zweig (2013) Linguistic Regularities in Continuous Space Word Representations

[9]Erik F. Tjong Kim Sang, Fien De Meulder (2003) Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition, CONLL '03 Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, volum 4, 142-147

[10] GuoDong Zhou, Jian Su (2002) Named Entity Recognition using an HMM-based Chunk Tagger, ACL '02 Proceedings of the 40th Annual Meeting on Association for Computational Linguistics,473-480

APÈNDIX

A aquest apèndix s'exposarà de forma detallada el funcionament del NER basat en probabilitats.

A1. FASE 1

Primerament creem dues variables per el train i pel test, aquestes variables venen donades per els fitxers "traint.txt" i "test.txt".

A2. FASE 2

Creem el nostre NER-arr a partir del training, NER_array es un array on s'exposa la quantitat de B_PER, B_ORG, B_LOC, B_MISC, I_PER, I_ORG, I_LOC i I_MISC del nostre train.
Per realitzar aquesta tasca primerament crearem el nostre lexicon que consisteix en un hasheable on esta cada paraula del text amb el seu NER i les vegades que apareix en el text. Per exemple:
"Marc va anar a la casa de Jaume perquè Jaume li va dir que li portés un llibre de la llibreria Figuerola."
En la frase anterior tindríem Marc com a B-PER, Jaume com a B-PER, Figuerola com a B-ORG i la resta de paraules com a O. El lexicon de l'anterior exemple seria el següent:

{ ('Marc', 'B-PER') : 1, ('va' , 'O') : 2, ('anar','O') : 1, ('a','O') : 1, ('la',O) : 2, ('casa','O') : 1, ('de','O') : 2, ('Jaume','B-PER') : 2, ('perquè','O') : 1, ('li','O') : 2, ('dir','O') : 1, ('que','O') : 1, ('portés','O') : 1, ('un','O') : 1, ('llibre','O') : 1, ('la','O') : 1, ('llibreria','O') : 1, ('Figuerola','B-ORG') : 1}

Un cop tenim el nostre lexicon creat mirarem la quantitat de paraules que apareixen una única vegada, a continuació introduirem al lexicon aquesta informació tractant aquestes paraules com desconegudes (unknown) en el exemple anterior per introduir per exemple la quantitat de B-PER que han aparescut posaríem *lexicon[("UNK", "B-PER")] = B_PER_ONCE* on B_PER_ONCE en el cas anterior seria igual a 1.
Un cop hem introduït els unknown en el nostre lexicon extraiem la quantitat de B-PER, I-PER, B-ORG, I-ORG, B-LOC, I-LOC, B-MISC, I-MISC i O del nostre taining i el retornem com un array, en aquest array es tenen en compte la quantitat original que hi havia de cada una de les etiquetes i les que hem introduït com a unknown.

Les funcions cridades en el codi son les següents:

```
X=hashCounts(train)
Y=insertUnknowns(X)
NER_arr = countNER(Y)
```

A3. FASE 3

Creem tranNER que consisteix en una llista on extraiem la quantitat de etiquetes de transició, es a dir, la quantitat de vegades que apareix una etiqueta després de la successió d'una altre. Les paraules d'inici de cada frase no queden registrades en el recompte perquè l'ultima paraula de la frase anterior no te perquè influir en l'etiqueta de la següent.
Per realitzar aquesta tasca primerament contem les transicions que apareixen en cada frase segmentant el train per frases i després fent el recompte, un possible output de la funció tindria aquest aspecte:

{('B-MISC', 'I-MISC'): 3, ('B-LOC', 'O'): 2, ('O', 'B-PER'): 4, ('I-PER', 'O'): 3, ('I-LOC', 'O'): 1, ('B-ORG', 'I-ORG'): 3, ('O', 'B-MISC'): 2, ('B-ORG', 'O'): 15, ('B-LOC', 'I-LOC'): 1, ('I-MISC', 'O'): 1, ('O', 'O'): 115, ('O', 'B-ORG'): 9, ('B-PER', 'I-PER'): 4, ('B-MISC', 'O'): 1, ('B-PER', 'O'): 3, ('O', 'B-LOC'): 2, ('I-ORG', 'O'): 3}

Un cop fet això l'únic que tenim que fer es un sumatori de cada "segona etiqueta" i posar-lo en una llista, la llista creada a partir del output anterior seria aquest:

[('B_PER', 4), ('B_ORG', 9), ('B_LOC', 2), ('B_MISC', 2), ('I_PER', 4), ('I_ORG', 3), ('I_LOC', 1), ('I_MISC', 3), ('O', 144)]

Les funcions cridades en el codi son les següents:

```
X=hashTransitionalCounts(train)
Tran_NER=countNER(X)
```

A4. FASE 4

El següent que extraurem serà lexical_prob que consisteix en un diccionari on apareixerà cada paraula del nostre lexicon amb les seves etiquetes i la probabilitat de tenir aquesta etiqueta.
Primerament extraurem el lexicon del train aconseguint cada paraula amb la seva etiqueta i les vegades que apareix en el train, a continuació insertarem els unknowns dintre del nostre lexicon.
Un cop tenim el nostre lexicon calcularem la probabilitat que te cada paraula en funció a la seva etiqueta a partir de les vegades que apareix etiquetada amb la seva etiqueta corresponent fent la divisió entre les vegades que apareixen i la quantitat total d'aquesta etiqueta en el train, aconseguit així un hash de (paraula,etiqueta) com a key i probabilitat com a valor.
Un cop tenim això creem lexical_prob on apareixerà cada paraula com a clau amb un hash de (etiqueta, probabilitat) com a valor.

Les funcions cridades en el codi són les següents:

```
X=hashCounts(train)
Y=insertUnknowns(X)
Z=hashLexicalProbs(NER_arr, Y)
Lexical_prob = conversionFunction(Z)
```

A5. FASE 5

El següent que extraurem serà transitional_prob que es una estructura igual que lexical_prob però en comtes de ferla a partir del nostre lexicon amb els seus unknowns es fara a partir de les paraules en transició.

Per realitzar aquesta tasca primerament contem les transicions que apareixen en cada frase segmentant el train per frases i després fent el recompte. A continuació aconseguirem les probabilitats de les transicions a partir de les transicions extretes i de tran_NER (variable extreta en un pas anterior) un exemple del que aconseguirem es el següent:

{('B-ORG', 'I-ORG'): 1.0, ('I-ORG', 'O'): 0.020833333333333332, ('O', 'B-ORG'): 1.0, ('I-PER', 'O'): 0.020833333333333332, ('O', 'B-LOC'): 1.0, ('O', 'B-PER'): 1.0, ('O', 'O'): 0.7986111111111112, ('I-MISC', 'O'): 0.006944444444444444, ('B-MISC', 'I-MISC'): 1.0, ('B-MISC', 'O'): 0.006944444444444444, ('B-ORG', 'O'): 0.10416666666666667, ('I-LOC', 'O'): 0.006944444444444444, ('B-LOC', 'O'): 0.013888888888888888, ('O', 'B-MISC'): 1.0, ('B-PER', 'I-PER'): 1.0, ('B-LOC', 'I-LOC'): 1.0, ('B-PER', 'O'): 0.020833333333333332}

Com podem observar la probabilitat de ('B-ORG','I-ORG') es de 1.0, això es una obvietat ja que sempre de I-ORG ha d'estar B-ORG.
Un cop tenim això creem transitional_prob on apareixerà cada paraula com a clau amb un hash de (etiqueta, probabilitat) com a valor. Un exemple de transitional_prob es el següent:

{'B-MISC': {'I-MISC': 1.0, 'O': 0.006944444444444444}, 'B-ORG': {'I-ORG': 1.0, 'O': 0.10416666666666667}, 'B-PER': {'I-PER': 1.0, 'O': 0.020833333333333332}, 'I-LOC': {'O': 0.006944444444444444}, 'I-ORG': {'O': 0.020833333333333332}, 'I-MISC': {'O': 0.006944444444444444}, 'B-LOC': {'O': 0.013888888888888888, 'I-LOC': 1.0}, 'O': {'B-MISC': 1.0, 'B-ORG': 1.0, 'B-PER': 1.0, 'B-LOC': 1.0, 'O': 0.7986111111111112}, 'I-PER': {'O': 0.020833333333333332}}

Els prints que s'han anat veient en l'explicació de com funciona el HMM implementat són de un fitxer training reduït, ja que, si volem mostrar resultats del training original es triga uns 15 o 20 minuts per print.

Les funcions cridades en el codi són les següents:

X=hashTransitionalCounts(train)
Y=hashTransitionalProbs(X)
Transitional_prob=conversionFunction(Y)

A6. FASE 6

A continuació aplicarem smoothing a la nostra variable transitional_prob, bàsicament en el que consisteix aquesta tasca es en revisar si B-PER, I-PER, B-LOC, I-LOC, B-ORG, I-ORG, B-MISC, I-MISC i O, afegint les que no estiguin posant com a probabilitat 0.0000001. Un cop fet això el nostre transitional_prob tindrà aquest aspecte:

{'I-LOC': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.006944444444444444, 'B-PER': 1e-07}, 'I-PER': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.020833333333333332, 'B-PER': 1e-07}, 'B-ORG': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1.0, 'I-MISC': 1e-06, 'O': 0.10416666666666667, 'B-PER': 1e-07}, 'B-LOC': {'I-LOC': 1.0, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.013888888888888888, 'B-PER': 1e-07}, 'B-MISC': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-MISC': 1.0, 'O': 0.006944444444444444, 'I-ORG': 1e-07, 'B-PER': 1e-07}, 'I-ORG': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.020833333333333332, 'B-PER': 1e-07}, 'I-MISC': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.006944444444444444, 'B-PER': 1e-07}, 'O': {'I-LOC': 1e-07, 'I-PER': 1e-07, 'B-ORG': 1.0, 'B-LOC': 1.0, 'B-PER': 1.0, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.7986111111111112, 'B-MISC': 1.0}, 'B-PER': {'I-LOC': 1e-07, 'I-PER': 1.0, 'B-ORG': 1e-07, 'B-LOC': 1e-07, 'B-MISC': 1e-07, 'I-ORG': 1e-07, 'I-MISC': 1e-06, 'O': 0.020833333333333332, 'B-PER': 1e-07}}

A7. FASE 7

Ja tenim la suficient informació per extreure els NER del nostre test, actualment tenim el següent:

- NER-array: Array amb els NER i les vegades que apareixen en el training incloent les paraules desconegudes.
- Tran_NER: Array amb els NER i les vegades que s'ha transicionat a aquestes en el training.
- Lexical_prob: Diccionari on apareixerà cada paraula del nostre lexicon amb les seves etiquetes i la probabilitat que te de tenir aquesta etiqueta.
- Tansitional_prob: Diccionari on apareixerà cada etiqueta amb la probabilitat de que sigui la predecessora de qualsevol altre etiqueta.

El primer que es realitza es contar quantes etiquetes i quines etiquetes apareixen al inici de cada frase en el nostre traint i les guardem en un diccionari, un exemple seria:

{'B-MISC': 2, 'B-ORG': 9, 'B-PER': 3, 'B-LOC': 1, 'O': 4}

A continuació extraiem les possibilitats de començament de cada paraula i el guardem en un diccionari que anomenarem NER_start_prob, un exemple seria:

{'B-PER': 0.15789473684210525, 'B-ORG': 0.47368421052631576, 'B-LOC': 0.05263157894736842, 'B-MISC': 0.10526315789473684, 'O': 0.21052631578947367}

Les funcions cridades en el codi són les següents:

X=hashSentStartCounts()
NER_start_prob=hashSentStartProbs(X)

Un cop tenim això creem una llista buida anomenada "pos" i comencem a llegir el test linea a linea, cada cop que llegim una paraula del test crearem una variable que es dirà "word" si la paraula del text esta en lexical_prob la variable "word" agafarà el seu valor, en cas contrari el valor de "word" serà "UNK". Si estem al inici de una frase recorrerem les keys de NER_start_prob i guardem a pos el l'etiqueta mes probable. En el cas de que la paraula no sigui el inici d'una frase recorrerem les etiquetes de l'anterior paraula analitzada, si aquesta es troba en lexical_prob es calcularà el score amb la següent formula:

score =(transitional_prob[prevNER][ner] * lexical_prob[word][ner])

A continuació s'afegirà a "pos" l'etiqueta mes probable. Un cop fet això tindrem el nostre test etiquetat. Per aconseguir un fitxer per validar el nostre NER basat en l'algorisme HMM anirem guardant en un fitxer anomenat output.txt la següent informació:

-Paraula tractada.
-Etiqueta correcta.
-Etiqueta predita.

Un cop fet això podrem veure els resultats mitjançant el script d'avaluació proporcionat a la conll 2002.

Un exemple de la llista "pos" seria la següent:

['B-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-ORG', 'I-ORG', 'B-ORG', 'I-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-ORG', 'I-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'B-ORG', 'I-ORG', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'O', 'B-ORG', 'I-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-ORG', 'I-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-ORG', 'I-ORG', 'B-ORG', 'I-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O', 'B-ORG', 'I-ORG', 'O', 'B-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'O', 'O', 'B-LOC', 'O', 'O', 'B-ORG', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'I-LOC', 'O', 'B-LOC', 'O']

Aquest exemple ha estat realitzat amb un fragment del traint i un fragment del test per no trigar excessivament en els temps d'execució.