Opinion Mining a Twitter

Albert Algilaga Santmiquel

07/02/2017

Resum– Avui dia la societat aprofita les xarxes socials com a mitjà d'expressió lliure, és per això que aquest document pretén fer un anàlisi dels sentiments que hi ha darrere de les opinions envers un tòpic com ara el Futbol Club Barcelona i el Reial Madrid. Per tal de fer-ho, s'entra en contacte amb la tecnologia que hi ha darrere d'aquest anàlisi, des de l'obtenció dels missatges utilitzats fins a l'extracció de les conclusions a partir dels seus sentiments. Els resultats de l'anàlisi d'aquesta mostra semblen afirmar que la societat tendeix a expresar-se més a favor d'un equip. Com a futures millores d'aquest projecte es podrien utilitzar xarxes neuronals amb un volum de dades més gran augmentant així l'efectivitat.

Paraules clau– Anàlisi, missatges, opinions, sentiments, societat, tecnologia, xarxes socials.

Abstract– Nowadays society uses social media as a way of expressing itself freely. For this reason, the aim of this paper is to analyze the feelings behind the opinions about a topic such as Futbol Club Barcelona and Real Madrid. In order to do it, it was got in touch with the technology behind this analysis from the utilized messages obtention to the conclusions extraction through their feelings. The results of this sample seem to suggest society is most likely to express in favor of a team. As a future improvements of this project neural networks could be used with more data in order to increase its efectivity.

Keywords– Analysis, feelings, messages, opinions, social media, society, technology.

1 Introducció

DURANT els últims anys les xarxes socials han suposat un medi d'expressió lliure on cadascú hi diu la seva opinió amb sinceritat aprofitant la protecció que ofereix estar al darrere d'una pantalla. Degut a això, ha sorgit la possibilitat d'examinar els sentiments de la societat per poder fer des d'estudis de mercat fins a obtenir orientacions polítiques segons el territori. Aquest Treball de Fi de Grau té com a objectiu analitzar els sentiments i opinions de la xarxa social Twitter. S'ha triat aquesta xarxa social per la facilitat a l'hora d'obtenir les piulades dels usuaris mitjançant la seva API i també perquè gran part de les piulades contenen etiquetes o hashtags en anglès. El tema que es tracta és si la societat de Twitter piula més a favor del Futbol Club Barcelona o del Reial Madrid.

• E-mail de contacte: albertalgilaga@gmail.com
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Jordi Casas Roma (dEIC)
• Curs 2016/17

2 Objectius

L'objectiu final d'aquest treball és poder expressar amb material visual la polarització de la societat envers un tema escollit i poder-ne treure conclusions. En el següent subapartat es detallen els subobjectius que componen l'objectiu final per ordre de realització.

2.1 Subobjectius

Per poder completar l'objectiu final, el projecte es divideix en les següents activitats:

- Obtenir piulades: Les piulades o 'tweets' s'obtenen gràcies a l'API de Twitter. Hi ha l'opció de recuperar tweets de dies anteriors indicant uns criteris de cerca dins un període de temps especificat o bé obtenir els tweets a temps real (streaming). S'ha optat per obtenir les piulades per streaming i poder extreure un gran volum de dades per arribar a una conclusió més encertada. Per filtrar les piulades s'han utilitzat com a hashtags o etiquetes preferents per valorar el sentiment barcelonista o madridista seran #ForçaBarça i #HalaMadrid de caire positiu i #Farça i #HalaMandril de caire

negatiu.

- Parsejar piulades: Els algorismes de Machine Learning necessiten uns inputs específics. Per tant, si els tweets no estan en el format desitjat, s'hauran de filtrar i netejar per tal de fer-ne un input vàlid per l'algorisme. Aquest procés transforma les piulades recaptades per després poder ser utilitzades.

- Classificar piulades: Aquesta activitat s'encarrega de classificar els tweets en positius i negatius. S'agafen les piulades tractades pel procés anterior i s'analitzen. Per poder aconseguir aquest subobjectiu s'utilitza un algorisme de Machine Learning supervisat, l'algorisme de classificació Naive Bayes. S'han utilitzat dues versions d'aquest algorisme: Naive Bayes basat en la llibreria Natural Language ToolKit (NLTK) i Naive Bayes propi, amb tractament de llanguatge natural propi. L'utilització d'aquestes dues versions té l'objectiu de veure si es podia equiparar l'efectivitat del Naive Bayes basat en NLTK amb el propi.

- Visualitzar de les dades: És important poder mostrar i expressar els resultats que s'han obtingut gràcies a les activitats anteriors de forma visual de manera que es puguin extreure conclusions ràpidament. Aquest punt crea sis gràfics diferents que s'ha cregut que són importants per extreure conclusions relacionades en el món del futbol:

  - Gràfic de dues barres que mostra el número de piulades queparlen sobre el Barça i el Madrid. L'eix d'ordenades representa la quantitat de tweets i l'eix d'abscisses representa els equips.
  - Gràfic de barres dobles per mostrar elnúmero de piulades negatives i positives que té cada equip. L'eix de les ordenades representa la quantitat de tweets (aquest eix està representat amb valors logarítmics ja que la diferència de entre el número de piulades positives i negatives és de la magnitud de 10^4) i l'eix d'abscisses representa els equips.
  - Gràfic lineal on es mostra la quantitat de tweets positius i negatius en el temps. En aquest cas l'eix d'ordenades també es representa amb valors logarítmics pel mateix motiu plantejat a l'anterior punt. En total hi ha dos gràfics d'aquest estil: un per representar els tweets del Barça i un per representar els tweets del Madrid. L'eix d'ordenades representa la quantitat de tweets, l'eix d'abscisses representen els dies on es van crear les piulades.
  - Gràfic lineal similar a l'anterior, però representant els tweets enla seva totalitat sense discriminar entre positius o negatius. Els eixos del gràfic representen el mateix tipus de dades que el gràfic anterior.
  - Gràfic lineal que representa el número de tweets en el temps per equips. Mostrant així quin és l'equip del qual s'ha parlat més durant aquests dies i quins han estat els dies que hi ha hagut més piulades per un equip i per l'altre.
  - Mapamundi on es representa amb punts blaus la localització de les piulades. Aquest gràfic ha estat possible realitzar-lo gràcies a l'atribut $created\_at$ que ens proporciona l'API de Twitter a l'enviar-nos les piulades en format JSON.

Els detalls tècnics sobre les tecnologies esmentades s'exposen en el punt 4. El procediment de classificació de les piulades s'explica pas a pas en el punt 5.2.

3 METODOLOGIA DE DESENVOLUPAMENT I PLANIFICACIÓ

Per dur a terme els objectius d'aquest projecte, s'ha utilitzat la metodologia de desenvolupament en cascada. S'ha triat aquesta metodologia perquè aquest projecte ha tingut unes dates d'entrega i uns objectius fixats des del primer moment. També cal destacar que, al ser un equip format per un component, resulta més efectiu seguir aquesta organització. De forma general, s'han complert els plaços dins els terminis establerts a la planificació i s'han entregat els informes dins els períodes fixats a l'aplicatiu web de l'assignatura.

El projecte es va començar el dia 3 d'Octubre (sense comptar els recursos temporals de la planificació, just després de l'entrega de l'Informe Inicial), la planificació seguida és la següent:

- Setama 9, 31 Octubre - 6 Novembre: Entrega de l'Informe de Progrés I. A aquestes dates les activitats d'obtenció de piulades i el filtratge de "tweets"han estat completades. A la vegada, ja s'ha començat a fer la recerca d'informació i llibreries per dur a terme l'activitat de Machine Learning.

- Setmana 15, 12 - 18 Desembre: Entrega de l'Informe de Progrés II. A la data de 18 de Desembre, l'activitat de Machine Learning i classificació de piulades ha estat completada i l'activitat de Visualització de Dades està en progrés, aquesta activitat es completaria el dia 27 de Desembre.

Si es contrasta la planificació inicial de l'apèndix amb les dates exposades en els anteriors dos punts, podem veure que s'ha seguit en la major part. L'única data que no s'ha respectat ha sigut el dia 12 de Desembre, on estava planificat tenir tota la part de programació i desenvolupament del l'analitzador de piulades acabada. Aquest retard en la finalització és degut a que l'activitat de Machine Learning i Classificació de piulades ha comportat un ús major de recursos temporals dels que estaven estimats pel sorgiment de problemes no previstos exposats al punt 4.

Per a més detalls sobre la planificaió inicial, veure la Fig. 15 de l'apèndix.

4 ESTAT DE L'ART I TECNOLOGIES UTILITZADES

4.1 Estat de l'art i relacions amb el projecte desenvolupat

Segons s'ha pogut trobar durant la recerca de quin algorisme utilitzar, la Universitat de Stanford, California ha desenvolupat un analitzador de comentaris que pot arribar

a un 85% d'encert en les prediccions de sentiments [3] que permet classificar des de 'Molt negatiu, Negatiu, Una mica negatiu, Neutral, Una mica positiu, Positiu i Molt positiu'. El co-fundador de Coursera, Andrew Ng, n'és un dels desenvolupadors. Aquest sistema utilitza una xarxa neuronal que permet llenguatge natural com a input i ha estat desenvolupada per treballadors de Google.

Analitzadors avançats com el que s'ha descrit en el paràgraf anterior estan entrenats amb milers de frases, que a la vegada s'han partit en n-grames, que són grups de paraules on n és la quantitat de items que hi ha per grup, per tant, un bi-gram és un grup de dues paraules. Aquest entrenament per grups de paraules permet una major precisió a la hora de decidir si una frase és positiva o negativa, ja que posseeix major informació que una sola paraula.

Tot i així, hi ha paraules que no són d'ajuda pels algorismes de classificació, les paraules d'un sol caràcter. La majoria d'analitzadors obvien aquestes paraules i no les tenen en compte a l'hora d'entrenar l'algorisme i d'analitzar els inputs. De la mateixa manera, els símbols de puntuació també s'eliminen i no es tenen en compte ja que no solen aportar un pes positiu o negatiu al text. Tot i així, hi ha analitzadors que aprofiten els signes d'exclamació i interrogació per emfatitzar els resultats, és a dir, si un classificador ha donat un output 'Negatiu', a l'haver-hi una exclamació a l'input, l'output es convertirà en 'Molt Negatiu'.

La idea general d'entrenament en el text mining és obtenir petits grups de paraules (n-grams) i, segons el sentiment de la frase en la que estaven aquests n-grams, incrementar el número de vegades que ha aparegut aquest n-gram en una frase positiva o negativa.

El 85% d'encert és un avanç molt important en aquest camp comptant que el màxim assolit anteriorment era un 80%. Però el que es busca també és la rapidesa d'implementació, d'entrenament, i d'execució. Aquest és el motiu pel qual s'ha triat l'algorisme Naive Bayes per classificar els tweets en aquest Treball de Fi de Grau. Naive Bayes funciona sorprenentment bé amb un petit volum de dades d'entrenament [4]. Per tant, si es té menys data set d'entrenament, es tarda menys en entrenar i al no ser tant complex d'implementar com les xarxes neuronals, es poden processar més inputs en menys temps. Si bé les xarxes neuronals es poden reentrenar (actualitzar l'entrenament previ i, per tant, no tornar a processar tot el data set d'entrenament), sol no tenir tanta eficàcia com tornar a entrenar de nou [2].

4.2 Tecnologies utilitzades

Aquest projecte s'ha realitzat utilitzant el llenguatge de programació Python v3.5, s'ha utilitzat aquesta versió perquè és l'últim *release* estable. Com a llibreries que no venen per defecte s'han utilitzat:

- Twitter-1.17.1: Llibreria que proveeix Twitter i que permet captar piulades en streaming. Aquesta proveeix les funcions Oauth, Twitter i TwitterStream que són necessàries per connectar amb el servidor de Twitter des d'on s'envien les piulades. Per realitzar la connexió és necessari tenir un compte de Twitter i crear un perfil de desenvolupador i crear una App. Aquest procés és necessari per poder obtenir l'ACCES TOKEN, l'ACCESS SECRET, la CONSUMER KEY i el CONSUMER SECRET per poder connectar l'analitzador de tweets amb Twitter pròpiament mitjançant OAuth. TwitterStream permet captar en streaming passant com a paràmetre els hashtags que interessen i l'idioma que es vol prioritzar. El següent fragment de codi mostra com crear la connexió a l'API de Twitter filtrant per hashtags ($\#ForçaBarça$, $\#Farça$, $\#HalaMadrid$, $\#HalaMandril$) i per idioma, en aquest cas castellà 'es'.

```
oauth = OAuth(ACCESS_TOKEN, ACCES_SECRET
    , CONSUMER_KEY, CONSUMER_SECRET)
#Per iniciar sessio
twitter_stream = TwitterStream(auth=
    oauth)

#Per obtenir tweets
hashtag = "#ForcaBarca,#Farca,#
    HalaMadrid,#HalaMandril"
print("Obtenint_piulades...")
iterator = twitter_stream.statuses.
    filter(track=hashtag, language='es')
```

- PyMongo: Llibreria que permet manipular la Base de Dades MongoDB des de l'analitzador que s'ha desenvolupat. Un cop inclosa, mitjançant la funció $MongoClient()$, es crea un client que es pot connectar al servidor de MongoDB, en aquest cas local, al port 27017.

- Cx_Oracle: Llibreria que permet manipular dades del servidor de SQL Developer (en aquest cas local). Per establir la connexió s'ha d'utilitzar la funció $cx\_Oracle.connect('USER/PASSWORD@IP')$, aquesta retorna la connexió reatiltzada. Un cop connectats a la BD relacional, es poden fer comandes SQL mitjançant la funció execute: $conexio.cursor\_BD.execute(query)$

- NumPy: Aquesta llibreria és necessària per utilitzar algunes funcions de matplotlib, per exemple, crear arrays de números seqüencials per poder crear gràfics.

- Matplotlib: La llibreria matplotlib permet la creació de gràfics en el llenguatge Python.

- TextBlob: Llibreria que permet utilitzar l'algorisme Naive Bayes basat en NLTK. Permet entrenar l'algorisme passant per paràmetre un array de frases polaritzades manualment, reentrenar l'algorisme un cop entrenat per primer cop, classificar frases i mostrar l'eficiència entre altres funcionalitats que no s'han utilitzat en aquest projecte.

Com s'ha dit anteriorment, aquest projecte utilitza una BD No SQL i una Base de Dades Relacional:

- Com a BD No SQL s'ha utilitzat MongoDB perquè és de lliure ús i emmagatzema dades en format JSON de forma dinàmica, és a dir, les entrades no tenen perquè tenir els mateixos atributs, sinó que es poden eliminar

atributs d'unes entrades determinades i les altres conservar aquests atributs [6]. Aquesta propietat és molt útil ja que hi ha tweets truncats o mal enviats que no tenen la majoria d'atributs que els tweets normals. L'atribut COORDINATES, quan està buit, té el valor 'None' i quan està ple guarda unes coordenades de tipus float.

- Com a BD relacional s'ha optat per utilitzar SQL Developer d'Oracle. Aquest entorn de desenvolupament de bases de dades a part de ser un entorn potent, és el que sempre s'ha utilitzat durant el grau i és el que es té més per mà a aquestes altures. S'ha instal·lat un servidor 11g per poder emmagatzemar les piulades un cop tractades.

Durant el desenvolupament del projecte s'han utilitzat tecnologies que no han estat les definitives, ja que ocasionaven problemes. Aquests són:

- Localització inventada: En la primera etapa, a la part de Filtrar els Tweets, les coordenades que s'emmagatzemaven no eren les que proporcionava el JSON del tweet, sinó les coordenades obtingudes a partir del nom de la ciutat on vivia l'usuari. El problema es presenta des de que, en aquesta informació, cada usuari pot posar el que vulgui, com per exemple:

	LOCALITZACIO	COORDENADES
1	INTERNATIONAL	(53.029256, -2.19979100233224)
2	In your living room	(14.5553557, 121.0249747)

Fig. 1: Coordenades falses.

Fig. 2: Localització de My Mind. Segons GeoPy està a Maryland

Per obtenir aquestes coordenades s'utilitzava la llibreria GeoPy amb la qual, a partir del nom de la ciutat, es poden obtenir les coordenades decimals d'aquesta. En la figura 1 es pot veure que aquesta llibreria té les coordenades de In your living room i de Internacional, la figura 2 mostra la localització de My mind.
S'utilitzava aquest sistema d'obtenció de coordenades pel fet de que molt pocs usuaris habiliten l'opció de localització de piulades, per defecte deshabilitada. Després d'una setmana recaptant piulades i coincidint amb el clàssic Barça - Madrid, concretament 235722 piulades van ser registrades on només 357 piulades tenien la localització habilitada. Es va creure que amb aquestes piulades n'hi hauria suficient com per plasmar-les al mapamundi (pels altres gràfics s'utilitzaran totes les piulades, no només 357), per tant, no hi haurà pèrdua d'informació. Així doncs, aquest problema ha estat solventat.

- Emoticones: Actualment hi ha piulades que no arriben a passar el filtre del segon procés (Filtratge de Dades) per tenir emoticones en el text, en el nom d'usuari o en ambdós llocs. Si bé hi ha un filtratge on s'eliminen les emoticones més freqüents, hi ha emoticones que no passen aquest filtre ja que utilitzen altres uns rangs de codi UNICODE que no s'han pogut trobar per falta d'informació i són descartats. Els rangs que filtren emoticones amb èxit són:
  - U0001F600-U0001F64F: per les emotiones més corrents.
  - U0001F300-U0001F5FF: pels símbols.
  - U0001F680-U0001F6FF: pels cotxes i símbols de mapes.
  - U0001F1E0-U0001F1FF: banderes en iOS.

5 Funcionament de l'analitzador

5.1 Arquitectura

L'analitzador de sentiments es compon per quatre mòduls independents entre sí, podent-se executar sense importar l'estat dels altres mòduls. Per aconseguir aquesta independència, cada mòdul interacciona amb les Bases de Dades i emmagatzema allà el seu output de manera que un altre procés pugui fer servir aquestes dades modificades. Com s'ha dit anteriorment, aquest sistema utilitza una BD No SQL i una BD relacional. La interacció dels mòduls amb les Bases de Dades és la següent:

- Mòdul 1 (Obtenir piulades): Les dades obtingudes de l'API de Twitter estan en format JSON. Aquestes dades es guarden 'en cru' a la Base de Dades no relacional (BD NoSQL). En aquest cas MongoDB, per tal de tenir una copia de seguretat i poder reiniciar el procés sense haver de dependre de l'obtenció de piulades per part de l'API de Twitter.

- Mòdul 2 (Parsejar piulades): Aquí es recuperen els tweets emmagatzemats pel mòdul anterior, es realitza el procés de neteja de les dades i s'escriu el resultat a la BD relacional. S'interactua amb la BD No SQL i amb la BD relacional. Primer de tot es recuperen les piulades de la BD NoSQL i se n'extreuen els atributs que interessen, aquests són: ID_Tweet, Cos del tweet, Nom_Usuari, Localització, Coordenades i Data de creació.

- Mòdul 3 (Classificar el sentiment de les piulades): Si hi ha noves dades a la BD relacional, les agafa i realitza el procés de classificació de sentiment. Com que l'output d'aquest procés és modificar dos camps de la BD relacional, modifica les columnes del registre que fa falta i actualitza la BD relacional. Només interactua amb la BD relacional.

- Mòdul 4 (Visualització de Dades): Aquest procés només agafa les dades de la BD relacional que calen

i les utilitza per generar material visual per poder-ne extreure conclusions. Per tant, només interactua amb la BD relacional.

El sistema, doncs es pot descriure de la següent forma:

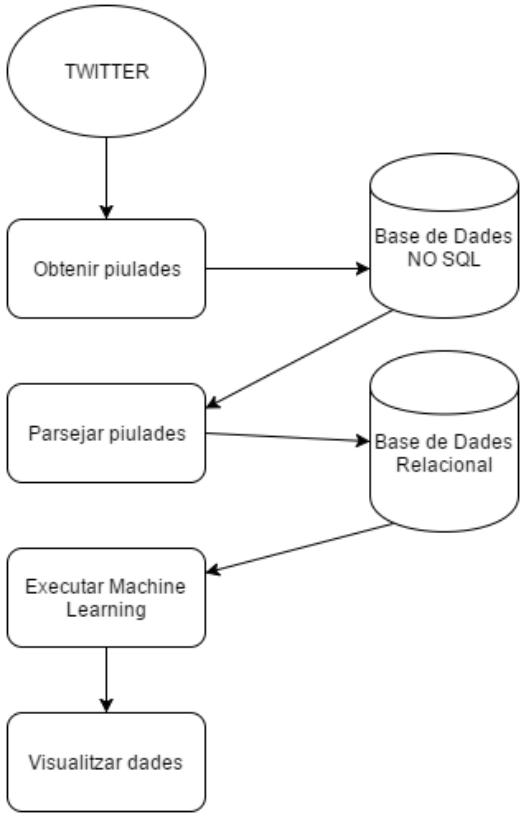

Fig. 3: Processos principals de l'analitzador

La BD relacional es compon d'una taula anomenada *TWEETS*, té les següents columnes:

- ID: Aquest camp és de tipus VARCHAR i emmagatzema l'ID del tweet obtingut pel camp '$\_id$' de la BD No SQL.
- TEXT: Aquest camp també és de tipus VARCHAR i permet guardar el cos del tweet, que es farà servir per analitzar la polaritat del missatge. Obtingut gracies al camp '$text$' de la BD No SQL.
- USUARI: Permet guardar el nom de l'usuari que ha creat el tweet. Obtingut del camp '$user : name$'.
- LOCALITZACIÓ: En aquest camp s'emmagatzema la localització de l'usuari. Cal recordar que aquesta localització no és d'on es crea el tweet. Obtingut gràcies al camp '$user : location$' de la BD No SQL.
- COORDENADES: Camp que guarda les coordenades en format [longitud, latitud]. Obtingut del camp '$coordinates$'.
- LLEGIT: Aquest camp pot tenir dos valors diferents. Quan té el valor de 1 significa que aquesta entrada ha estat analitzada, d'altra banda, quan val 0 està encara per tractar.
- POLARITZACIÓ: A l'igual que el camp anterior aquest camp pot adoptar dos valors, 1 i 0. Quan val 0 vol dir que la piulada és negativa, en canvi, quan val 1 vol dir que és positiva.
- DATA: Camp que permet saber quan ha estat creat el tweet. Aquest camp s'obté gràcies a l'atribut '$created\_at$' de la BD No SQL.

	COLUMN_NAME	DATA_TYPE
1	ID	VARCHAR2(30 BYTE)
2	TEXT	VARCHAR2(1120 BYTE)
3	USUARI	VARCHAR2(50 BYTE)
4	LOCALITZACIO	VARCHAR2(200 BYTE)
5	COORDENADES	VARCHAR2(200 BYTE)
6	LLEGIT	NUMBER(1,0)
7	POLARITZACIO	NUMBER(1,0)
8	DATA	VARCHAR2(200 BYTE)

Fig. 4: Columnes de la Base de Dades relacional

La BD No SQL emmagatzema tweets amb els següents camps útils: '$\_id$', '$text$', '$user : name$', '$user : location$', '$coordinates$' i '$created\_at$' explicats a l'anterior paràgraf. La resta de camps també s'emmagatzemen però no s'utilitzen.

5.2 Classificació d'una piulada: Pas a pas

L'analitzador propi d'aquest Treball de Fi de Grau utilitza l'algorisme Naive Bayes com a classificador de tweets, ja que aquest té un data set d'entrenament modest i un gran volum de tweets per analitzar. Com a output del classificador hi ha 'Positiu' o 'Negatiu'. Per tenir un data set de major qualitat, s'eliminen les paraules d'una sola lletra i els signes de puntuació, a la vegada també s'ignoren els noms propis dels jugadors del Barça i el Madrid per evitar relacionar-los amb outputs positius o negatius. Per acabar, es transformen totes les paraules en lletra minúscula per evitar més d'una entrada per paraula al diccionari de freqüències.

El Naive Bayes propi s'entrena creant un diccionari de paraules a mesura que llegeix el data set d'entrenament. En aquest cas de 288 entrades entre tweets agafats de la Base de Dades relacional i frases freqüents. A partir d'aquest data set d'entrenament (Train Set), es tokenitza paraula per paraula cada entrada. Cada paraula apareguda en el text es compta quantes vegades ha aparegut en un tweet positiu o negatiu, de manera que, després, es poden calcular la probabilitat de que la paraula sigui negativa o no. Després de calcular les probabilitats de cada paraula, s'omple o actualitza l'entrada del diccionari python que correspon a la paraula quedant amb el següent format:

key : value
$Paraula : [\#POS, \#NEG, PROB\_POS, PROB\_NEG]$

Per entrenar el Naive Bayes s'agafen piulades emmagatzemades a la BD Relacional i se n'eliminen les partícules no desitjades, ja que seria soroll dins el Train Set. Per netejar les piulades del Train Set s'eliminen les paraules formades per un sol caràcter, s'obvien les referències (mots precedits per una @, per exemple, $RT@Usuari$ o bé

$@Nom\_Usuari$) així com també els hashtags i els noms dels jugadors del Futbol Club Barcelona i del Reial Madrid.

En aquest punt s'exposa un cas pràctic de seguiment d'un tweet a través del procés d'anàlisi del seu sentiment. Es pren com a exemple un tweet amb les següents dades:

- $\_id$: 5842bd3591869f1144482ddc.
- $Text$: Preparado para el Clásico #ForçaBarça #FCBarcelona #Messi #BarçaVSMadrid #LaLiga #CampNou... https://t.co/phJTe9dsEL.
- $User : name$: Plácido.
- $User : location$: Valverde del Camino, España.
- $Coordinates$: [-6.75, 37.5667].
- $Created\_at$: Sat Dec 03 12:39:39 +0000 2016

En aquest punt representa que ha estat emmagatzemat a la BD No SQL a punt per ser tractat. El següent pas a seguir és emmagatzemar-lo a la BD relacional. Per fer això, cal primer que passi el filtratge per saber si és un bon candidat.

Primer de tot es mira si té el camp '$text$' i el camp '$user : name$' i es comprova que passen pel filtratge d'emoticones. Com que cap dels dos camps té emoticones, llavors es comprova que tinguin els camps '$\_id$', '$user : location$' i '$created\_at$'. De moment aquest tweet és un bon candidat i ha passat el filtratge, per tant, no és un tweet truncat (si es tractés d'un tweet truncat tindria l'atribut '$hangup$' amb valor '$True$' i no tindria cap dels atributs anteriors). D'aquesta manera ja es pot emmagatzemar a la BD reacional.

Cada columna descrita al punt 5.1 s'omplirà amb la dada que l'hi pertoca. A més dels atributs obtinguts de la BD No SQL, se n'afegeixen dos més: '$llegit$' amb valor 0 per defecte i '$polaritat$' també amb valor 0 per defecte. Ara aquest tweet ja està netejat i comprovat.

El següent pas a seguir és classificar el tweet. Un cop entrenat l'algorisme amb el fitxer d'entrenament omplert manualment i el diccionari de freqüències creat, ja es pot classificar el tweet. Es fa un *split* del text del tweet, és a dir, s'agafa paraula per paraula i es mira al diccionari de freqüències les probabilitats de cada paraula de ser positiva o negativa. La probabilitat de que una paraula sigui positiva es calcula seguint la següent fórmula:

$$(paraula|TweetPositiu) = \left(\frac{\#TweetPosParaula}{\#TweetsPositius}\right) \quad (1)$$

La possibilitat de que una paraula sigui positiva (o negativa) és igual a la divisió entre el número de piulades positives (o negatives) on apareix la paraula $\#TweetPosParaula$ i el número de piulades positives (o negatives) totals $\#TweetsPositius$.

De la mateixa manera, el càlcul de la negativitat d'una paraula es farà seguint la fórmula anterior canviant les probabilitats de positivitat amb els valors de negativitat.

Tornant al cas pràctic, l'analitzador calcula la positivitat i negativitat de cada paraula recuperant els valors del diccionari de freqüències:

- 'Preparado' no existeix al diccionari de freqüències i no es té en compte.
- 'para' té una negativitat de 1'01221 i una positivitat de 1'01137.
- 'el' té una negativitat de 1'03540 i una positivitat de 1'03640.
- 'clásico' no existeix al diccionari ja que es va entrenar amb tweets creats setmanes abans del clàssic i no en parlaven.
- La resta de paraules són hashtags i un link i que, com s'ha dit en punts anteriors, a la hora d'entrenar l'algorisme s'obvien i es descarten, per tant, no existeixen al diccionari.

Un cop obtingudes totes les probabilitats de ser positiva i negativa cada paraula, se segueix la següent fórmula per calcular la polarització del tweet:

$$(pol|p_1, \ldots, p_n) = P\,(pol) \prod_i^n (p_i|pol) \quad (2)$$

on pol indica la polaritat i p_i fa referència al conjunt de paraules que conformen la piulada.

Es multipliquen els valors de positivitat (o negativitat) de cada paraula que s'han calculat amb anterioritat $(paraula_i|polaritat)$ amb la probabilitat de que una paraula, de forma general, sigui positiva (o negativa) $P\,(polaritat)$.

Cal calcular les probabilitats de que cada paraula sigui positiva o negativa. Es calcula dividint el número de piulades positives o negatives entre el número total de piulades del Train Set. L'analitzador calcula que:
$P(POS) = 0.5176678445229682$
és la probabilitat de que una paraula sigui positiva segons el Train Set.
$P(NEG) = 0.4823321554770318$
és la probabilitat de que una paraula sigui negativa segons el Train Set.

Ara, seguint la fórmula anterior, es pot calcular la polaritat del tweet. Per exemple, calculem la positivitat:
$(positiu|Preparado, para, el, classico) = P(POS) * 1 * (para|TweetPositiu) * (el|TweetPositiu) * 1$
$0.51766 * 1'01137 * 1'03640 = 0.54261$

L'analitzador dóna el resultat final de:
positivitat tweet: 0.5426172261819098
negativitat tweet: 0.5055088871343288

Com que la positivitat és més gran que la negativitat del tweet, l'analitzador retorna el valor de 1 i modificarà la columna '$polarització$' de l'entrada de la piulada a la BD

relacional de 0 a 1.

Aquest, doncs, és el camí que segueix una piulada a l'hora de ser classificada, utilitzant la implementació pròpia. Té una certesa del 63'33%.

Quan s'utilitza la llibreria de NLTK es fa servir el mateix Train Set. Es passa per paràmetre un array amb les entrades del fitxer d'entrenament i la llibreria fa la resta. Per entrenar-lo només fa falta escriure la següent línia de codi:

```
classificador = NaiveBayesClassifier(train)
```

On *train* és l'array que conté el fitxer d'entrenament. Segons la documentació de la llibreria de TextBlob utiltza un procés semblant a l'implementació pròpia: elimina les paràules d'un sol caràcter i els signes de puntuació. Un apartat que difereix és que *tokenitza* el text en tri-grams, bi-grams i uni-grams segons el *tokenitzador* que utilitzi, pot ser $SentenceTokenizer$ des de bi-grams a tri-grams i $WordTokenizer$ que retorna uni-grams [9]. Aquesta implementació obté un 73'33% d'efectivitat, segurament pel fet de que les unitats de tractament estan formades per més d'una paraula.

6 INTERPRETACIÓ DELS RESULTATS

Observant els gràfics creats per l'analitzador es poden arribar a un conjunt de conclusions.

Si es dóna un cop d'ull a la figura 5 es pot veure que la majoria de tweets es van captar els dies on un dels dos equips o bé els dos jugaven un partit. Dins aquests dies hi ha 3 pics de recaptació de piulades: el dia 03 de desembre (Lliga: Barça – Madrid), 07 de desembre (Champions: Madrid – Borussia Dortmund), 10 de desembre (Lliga: Madrid – Deportivo; Lliga: Osasuna – Barça).

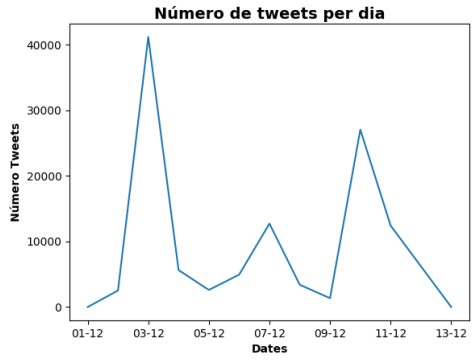

Fig. 5: Número de Tweets per dia

Concretament durant aquests dies es van captar un total de 235722 piulades. D'aquestes, 113900 van passar el filtratge i han estat emmagatzemades a la BD Relacional. Dins d'aquests tweets vàlids, 41217 (el 36'18%) es van recollir el dia del clàssic. En la figura 6 es pot veure clarament com l'estadi del Futbol Club Barcelona (Camp Nou) és el lloc on es van fer més piulades ja que n'hi ha una gran densitat.

Fig. 6: Mapa de Barcelona. Dins el cercle vermell es troba el Camp Nou, amb gran densitat de piulades.

S'ha dit que al Camp Nou hi ha una alta densitat de piulades, si es mira l'estadi del Reial Madrid (Santiago Bernabéu) també hi ha una gran densitat de piulades, de la mateixa manera, la Puerta del Sol de Madrid també hi ha una gran densitat (veure la figura 7). Un factor comú d'aquestes localitzacions és que són llocs emblemàtics, per tant, es podria arribar a la conclusió que als seguidors del Barça i el Madrid els agrada o pensen en activar la localització de Twitter quan hi ha algun succés important o bé estan creant una piulada en algun lloc emblemàtic.

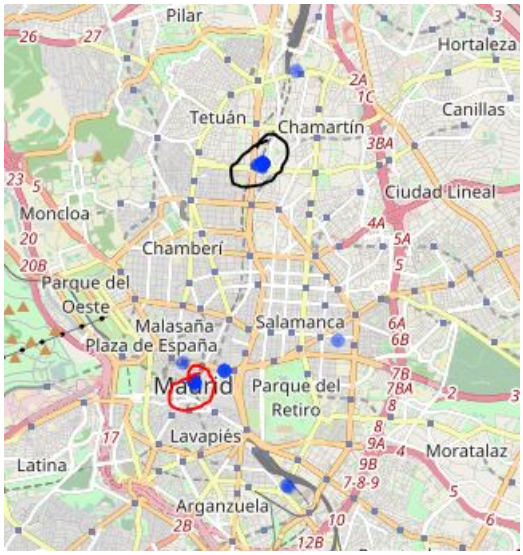

Fig. 7: Mapa de Madrid. Dins el cercle vermell es troba la Puerta del Sol, a dins el cercle negre es troba l'estadi Santiago Bernabéu. Els dos llocs amb alta densitat de piulades.

Degut al problema de les localitzacions falses descrit a l'apartat 4.2 s'han obtingut menys tweets amb la localització habilitada. Com s'ha comentat, moltes piulades han estat creades des del mateix punt del mapa aportant poca varietat territorial. Com a objectiu secundari, es volia saber la distribució barcelonista i madridista a nivell de territori català i espanyol, però no ha pogut ser possible ja que amb aquest número de piulades localitzades no es pot deduir un esquema o un patró d'on viuen els aficionats culers i madridistes. La figura 8 mostra aquest fet.

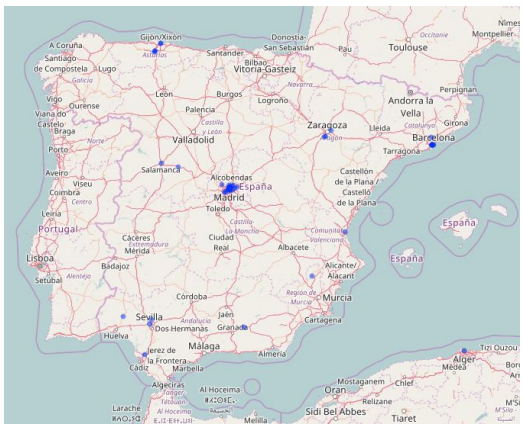

Fig. 8: Distribució de les piulades a l'Estat espanyol

Cal destacar que la gran majoria de piulades són positives segons la figura 9. D'aquí es pot extreure la conclusió que els internautes tendeixen a crear piulades en suport al seu equip més que criticar el seu equip, per exemple en cas de derrota, o l'equip rival. Com a apunt important caldria dir que moltes de les piulades positives són *retweets* de declaracions dels jugadors i premsa esportiva. Com que aquestes declaracions solen ser de neutrals a positives, augmenten la quantitat de piulades positives. Per exemple: RT @FCBarcelona_es: ¡Andrés Iniesta recibe el alta y ya está a punto para el clásico #ForçaBarça #ElClásico

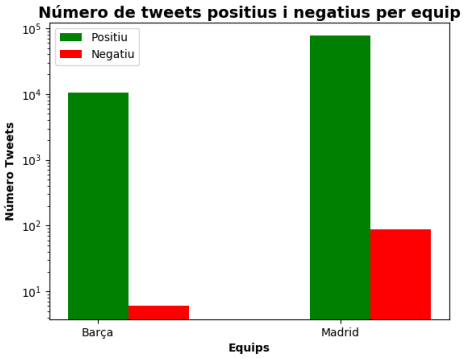

Fig. 9: Número de Tweets positius i negatius per equip

De la mateixa manera les figures 10 i 11 confirmen que hi ha un major volum de piulades de suport que de crítica negativa. Aquestes figures representen el nombre de piulades positives i negatives diàries. Es pot veure que en tot moment les piulades positives són majoria.

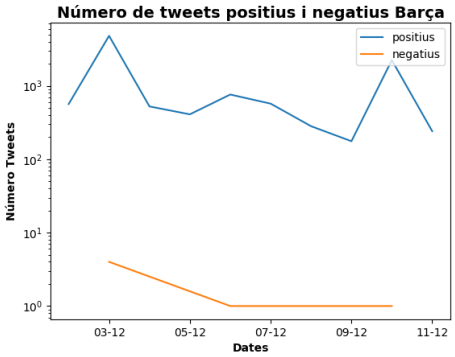

Fig. 10: Número de Tweets positius i negatius Barça

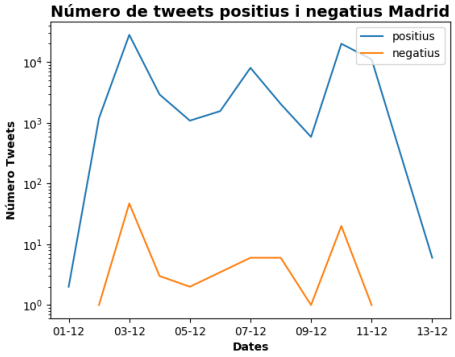

Fig. 11: Número de Tweets positius i negatius Madrid

En la figura 12, on es mostren les piulades localitzades, es pot veure que els punts estan concentrats majoritàriament a països de parla hispana i a Indonèsia. Això és degut a què només s'han captat piulades en castellà ja que així es va planejar a l'informe inicial. Indonèsia és un dels països on el Barça està molt present en la societat, és molt possible que els indonesis hagin fet retweet de tweets en castellà i/o que els immigrants de parla hispana hagin piulat des de Indonèsia.

Fig. 12: Localització dels tweets captats en el món

Amb les figures 13 i 14 es pot observar la diferència entre els dos equips segons les piulades que representen. S'han captat més piulades del Reial Madrid que del Futbol Club Barcelona, segurament perquè s'han filtrat només els tweets en castellà. Aquest succés pot tenir la lectura de que gran part dels seguidors del Barça són, segurament, de parla catalana. Si es filtressin tweets de parla catalana segurament els

gràfics tindrien uns resultats totalment inversos: El Barça tindria més piulades que el Madrid.

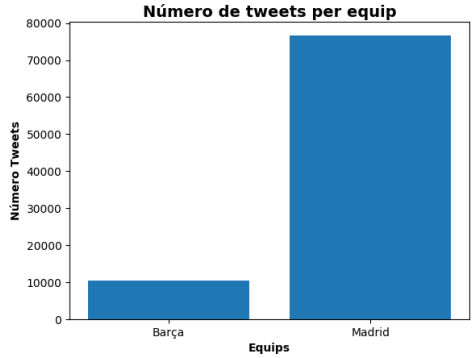

Fig. 13: Número de Tweets per equip

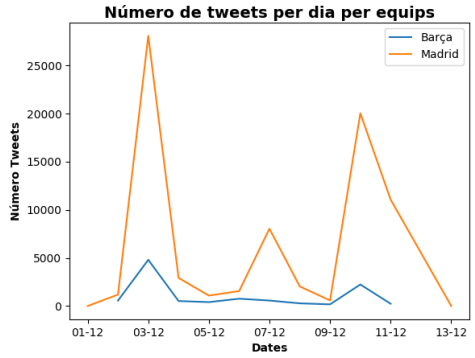

Fig. 14: Número de Tweets per dia per equips

Per tant, segons la informació que permeten extreure els gràfics, es pot dir que del 01 de desembre al 13 de desembre les persones que van fer piulades amb els hashtags descrits al punt 1 d'aquest article són més propenses a escriure piulades que parlen del Madrid i de forma positiva. En conclusió, aquests dies la societat de parla hispana de Twitter estava més a favor del Reial Madrid que del Futbol Club Barcelona.

7 Conclusions

Aquest treball s'ha dividit en quatre mòduls independents entre si: Obtenir dades, Filtrar Dades, Classificar Dades, Visualitzar Dades. El mòdul tercer ha estat el que ha comportat més recursos temporals, sobretot en recerca. Tal com s'ha dit, actualment l'algorisme de Machine Learning propi té un 63'33% d'efectivitat mentre que el basat en la llibreria NLTK, té un 73'33%. Una manera de poder augmentar aquesta efectivitat per aconseguir igualar l'efectivitat del propi amb l'altra implementació seria augmentar el número de piulades del Train Set, actualment 288, i utilitzar n-grams (més d'un item per token) en comptes de tokenitzar paraula per paraula a la hora de construir el diccionari per entrenar l'algorisme, ja que solen tenir un major rendiment.

De cara a millorar l'efectivitat de forma dràstica seria interessant utilitzar xarxes neuronals amb un Train Set amb una gran quantitat d'entrades, ja que les xarxes neuronals amb un gran volum d'entrenament solen funcionar molt millor que amb volums petits, com més entrenament, millor eficàcia [2]

Com a possibles millores d'aquest projecte: es podrien guardar les piulades en un sistema distribuït amb mirrors per assegurar la disponibilitat i augmentar la seguretat de l'emmagatzematge de les dades. Per exemple amb HDFS, el sistema distribuit de Hadoop. També seria interessant poder executar en màquines diferents cada mòdul, és a dir, una màquina que Obtingui piulades, una altra que dugui a terme el filtratge de les dades, una altra que polaritzi les piulades i una quarta que s'encarregui de generar els gràfics i material visual. Per tant, un sistema distribuït per fer funcionar aquest projecte d'Opinion Mining pot ser una bona aposta.

Agraïments

El meu més sincer agraïment al meu tutor Jordi Casas Roma per estar sempre disposat a donar un cop de mà i guiar-me durant la realització d'aquest projecte. També donar gràcies al dEIC per la disposició d'una màquina del departament. No voldria acabar sense agrair a la meva família i a la meva xicota el suport donat en aquest repte que se m'ha presentat. Gràcies.

Referències

[1] M. Bonzanini, (2015). Mining Twitter Data With Python. [En línia] Disponible: https://marcobonzanini.com/2015/03/02/mining-twitter-data- with-python-part-1/

[2] A. Farhangi, (2015). When should I use Naive Bayes over neural networks? [En línia] Disponible: https://www.quora.com/When-should-I-use-Naive-Bayes-classifier-over-neural-networks

[3] Knowingly, Inc., (2017) Standford researchers to open-source model they say has nailed sentimental analysis [En línia] Disponible: https://gigaom.com/2013/10/03/stanford-researchers-to-open-source-model-they-say-has-nailed-sentiment-analysis/

[4] G. Forman & I. Cohen, (Data no especificada). Learning from Little: Comparison of Classifiers Given Little Training [En línia] Disponible: http://www.ifp.illinois.edu/ iracohen/publications/precision-ecml04-ColorTR-final.pdf

[5] Twitter Inc., (2016). Twitter Apps. [En línia]. Disbonible: https://apps.twitter.com/

[6] MongoDB, Inc. (2017). MongoDB and MySQL Compared. [En línia] Disponible: https://www.mongodb.com/compare/mongodb-mysql

[7] Oracle Support, (2016). OBE and Tutorials. [En linia] Disponible: http://www.oracle.com/technetwork/developer-tools/sql-developer/obe-082749.html

[8] S. Raschka, (2014). Naive Bayes and Text Classification. [En linia] Disponible: http://sebastianraschka.com/Articles/2014_naive_bayes_1.html

[9] S. Loria, (2016) API Reference. [En linia] Disponible: https://textblob.readthedocs.io/en/dev/api_reference.html #textblob.base.BaseTokenizer.tokenize

APÈNDIX

A.1 Planificació del projecte: Diagrama de Gantt

1. Obtenir piulades
   (a) Accés a l'API
   (b) Obtenir piulades
   (c) Programar connexió amb la Base de Dades No SQL
   (d) Guardar piulades
2. Parsejar dades
   (a) Programar neteja piulades
   (b) Programar connexió amb Base de Dades Relacional
   (c) Guardar piulades netejades
3. Machine Learning
   (a) Recerca algorismes
   (b) Execució algorismes
4. Visualització dades
   (a) Recol·lectar resultats
   (b) Plasmar resultats en mapes i gràfics
   (c) Extreure conclusions

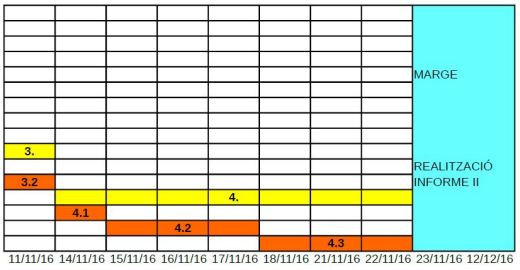

Fig. 15: Diagrama de Gantt