Opinion mining in Twitter: Anàlisi de sentiments o opinions

Cristian Puig Aribau

Resum– Actualment estem en l'era de la informació, la nostra societat cada cop necessita més una informació precisa i filtrada, que en molts casos, fins i tot, pot representar l'èxit o la fallida de moltes de les empreses actuals. És per aquest motiu que les xarxes socials suposen un nou canal d'informació, que tractant-la adequadament es poden transformar en resultats de gran valor. Al llarg d'aquest projecte s'ha aprofundit en l'adquisició de la informació mitjançant la xarxa social Twitter i s'ha realitzat una simulació amb dades reals, ponderant les emocions de les dades recollides i arribant en un resultat gràcies als algoritmes de processament del llenguatge natural i classificació de text. Així veient tots els passos a seguir per a poder extreure unes conclusions adients que ens permetin obtenir una globalització capaç de definir si un esdeveniment ha estat positiu, negatiu o neutre.

Paraules clau– Twitter, informació, MySQL, MongoDB, ETL, anàlisi, NLTK, API Prediction Google

Abstract– Nowadays we are in the information era, our society needs specific information to their needs, and in many times this represents the success or failure of a company. Is for this reason that the social media have become a very important information channel, which using it properly could help us to achieve very important goals. Throughout this project we got into the gathering information process by using the social media tool Twitter. We have done a simulation with real data pondering the emotions of the information gathered and getting a result thanks to the algorithm natural language processing and text categorization. This simulation helps us to understand the steps to follow to be able to get some conclusions, and decide at the end if an event has been positive, negative or neutral.

Keywords– Twitter, information, MySQL, MongoDB, ETL, analysis, NLTK, API Prediction Google

1 Introducció

Avui en dia, gairebé tota la societat o la major part d'ella utilitza les xarxes socials; ja sigui per comunicar-se amb altres usuaris, per expressar les seves opinions, sentiments, estats, o simplement per informar-se a temps real de què està succeint arreu del món. Durant 60 segons, a la xarxa, es comparteixen milions de dades, i aquestes dades no desapareixen; aquestes dades queden emmagatzemades en els grans servidors de les empreses que ens donen el servei, com pot ser Twitter.

Un fet molt important, és que inconscientment, escrivim i publiquem informació personal a les xarxes socials sense saber que, de totes aquestes dades se'n podran realitzar estudis i extreure resultats, i que empreses interessades pagaran per aquesta informació. Una de les xarxes socials més important i popular actualment és Twitter. Twitter, que va ser creada l'any 2006, és una xarxa social on els seus usuaris utilitzen 140 caràcters per a expressar una idea, un missatge o una oferta. Aquesta particularitat fa que Twitter es conegui com a una xarxa social de microblogging, a més a més amb la utilització de *"hashtags"* (fent servir el símbol "#"davant d'una paraula) es relacionen *tweets*, d'aquesta manera podem filtrar i veure només, informació d'aquell *"#hashtag"*.

Un cop introduït el concepte de xarxa social, i en concret Twitter, cal entrar en la base del projecte escollit. Aquest treball es basa en l'anàlisi de sentiments i opinions d'un tema utilitzant la xarxa social Twitter com a font d'informa-

• E-mail de contacte: crispuar@gmail.com
• Menció realitzada: Enginyeria de Tecnologies de la Informació
• Treball tutoritzat per: Jordi Casas Roma (DEIC)
• Curs 2015/16

ció, per tant, l'objectiu principal és veure, analitzar i treure conclusions d'un tema actiu a la xarxa social, veient així les diferents fases que s'han de passar per arribar a treure unes conclusions i resultats.

La motivació principal d'aquest projecte ve donada per l'interès i la curiositat en l'àmbit de l'anàlisi d'informació en xarxes socials, ja que avui en dia la societat es passa més hores compartint les seves vivències a la xarxa i aquestes poden ser estudiades. Per altra banda, per veure i investigar cadascun dels passos a seguir per a l'obtenció d'uns resultats finals. També cal dir que és un tipus de projecte que actualment es treballa molt, ja que en moltes campanyes de màrqueting o campanyes publicitàries es fan servir aquests estudis. Dit d'una altra manera, es fan estudis diàriament de diferents temes publicats en xarxes socials, en concret Twitter, i aquesta és la tasca que es vol portar a terme. Es pot dir que actualment estem en l'era de la informació digital i que les dades són el petroli del segle XXI.

1.1 Objectius

L'objectiu principal d'aquest projecte és veure els passos a seguir per arribar a uns resultats mitjançant un sistema que ens permeti l'anàlisi d'un tema escollit, per tal de poder veure si els usuaris de Twitter pensen positivament o negativament (el seu estat d'ànim) sobre un tema, així mostrant-ne la tendència. A més a més, s'analitzaran i es compararan alguns dels algoritmes de *Machine Learning* (processament del llenguatge natural i classificació de text) existents per a poder dur a terme aquesta tasca.

Per tal d'aconseguir aquest objectiu global, caldrà assolir els següents subobjectius:

- Implementar un mòdul per a l'obtenció i filtratge de dades de Twitter
- Implementar la Base de Dades per a guardar tota la informació recollida
- Implementar algoritmes de processament del llenguatge natural i processament de text
- Visualització i anàlisi dels resultats

A la Figura 1. Podem veure la relació que hi ha entre els 4 subobjectius

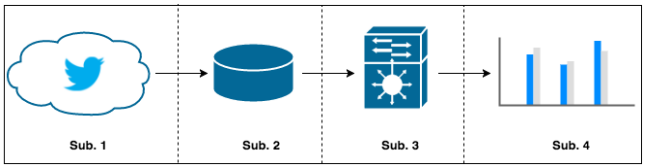

Fig. 1: Relació dels subobjectius

Aquest article està organitzat en els següents apartats:

En la secció 1, ja s'ha vist la introducció del projecte i els seus objectius. En la secció 2 presentarem l'estat de l'art, on descriurem estudis de dades de Twitter semblants a l'estudi a realitzar i la importància que pot arribar a tenir, també es comentarà l'ús de les bases de dades NoSQL en aquests tipus de projectes. A continuació, a la secció 3, s'explicarà la metodologia seguida per al desenvolupament del projecte, conjuntament amb les tecnologies utilitzades. Seguin amb la secció 4, s'exposarà la planificació i l'estructura del projecte. A la secció 5, entrarem en el desenvolupament i experiments del projecte. En aquest apartat es comentaran els passos que s'han seguit per arribar als resultats obtinguts. La utilització dels diferents mètodes per assolir l'objectiu principal i com funcionen els algoritmes escollits pel processament del llenguatge natural i classificació de text. Se seguirà en la secció 6, on presentarem els resultats obtinguts mitjançant gràfics i explicació, així com la tendència de les dades recollides. Per finalitzar a la secció 6, s'exposaran les conclusions obtingudes a través del desenvolupament i resultats del projecte.

2 Estat de l'art

Actualment hi ha moltes empreses que fan aquests tipus d'anàlisi per a satisfer diferents necessitats, ja siguin per a poder preveure fets pròxims o per estudiar la tendència d'un tipus de públic en concret, entre molts d'altres tipus. Abans de realitzar aquest projecte es va veure un exemple d'un cas similar. "WebSays"[1], és una empresa que es dedica a monitorar xarxes socials, i una d'elles és Twitter. Un dels estudis realitzats es va centrar en les passades eleccions del 20-D, on es va fer una estimació de l'ordre de popularitat de cada candidat abans que se celebressin les eleccions. Com es pot apreciar als resultats, l'estimació era força precisa. Per tant es pot dir que van realitzar una bona feina d'anàlisi i processament de text per arribar a extreure resultats que es plasmen a la vida real.

Existeixen diferents algoritmes de processament i classificació de text que segons un conjunt de dades ja classificades, donen uns resultats. Aquests algoritmes ja implementats donen un seguit d'opcions que permeten saber les tendències d'un tema o producte escollit. Google, com empresa pionera el sector informàtic, també té una API (una eina) que et permet fer aquest tipus d'anàlisis mitjançant peticions REST. Moltes empreses ja fan servir aquesta eina per a poder extreure els seus propis resultats.

A causa de l'excés d'informació digital que es genera diàriament per culpa de les xarxes socials, el fet que aquesta informació s'ha d'emmagatzemar en bases de dades i que l'estructura d'elles no pot canviar constantment; ha sorgit el concepte de les bases de dades NoSQL. Aquest tipus de bases de dades s'utilitzen molt en la pràctica de *"Big Data"* i *"Machine Learning"*, ja que poden emmagatzemar molta informació que no cal tenir estructurada. En els últims anys han tingut una ascensió molt forta en l'àmbit informàtic. Existeixen diferents bases de dades NoSQL: basades en clau/valor, en documents, en grafs, tabulars, entre d'altres. Segons el tipus de projecte que es vulgui porta a terme utilitzarem unes o altres.

3 METODOLOGIA

3.1 Metodologia utilitzada

La metodologia emprada per a portar a terme aquest projecte i aconseguir tots els objectius establerts ha estat la metodologia en cascada. És un tipus de metodologia iterativa, i per tal de porta-la a terme, cal que, des d'un bon principi, existeixin unes fases molt ben definides, ja que per avançar en una fase cal haver superat l'anterior, tot i que les tasques de desenvolupament van molt lligades entre elles. La decisió en la selecció d'aquesta metodologia ve donada pel temps que hi ha per a desenvolupar aquest projecte i els recursos disponibles, que són només d'una persona.

Les fases que especifica aquesta metodologia són les següents, i adaptades en aquest projecte queden de la següent manera:

- **Anàlisi i requeriments**: en aquesta fase s'ha realitzat un anàlisi del que es vol per tal que el projecte tingui els seus fruits.
- **Disseny**: en aquesta s'ha realitzat el disseny de les parts que ho necessiten, com les bases de dades.
- **Desenvolupament**: en aquesta fase s'ha implementat el codi de les diferents parts del projecte, ja que en aquest, es tenen diferents parts de desenvolupament molt separades.
- **Test i post anàlisi**: durant aquesta fase, s'han realitzat les proves pel bon funcionament de cada mòdul de desenvolupament segons requeriments i comparació de resultats. (Aquesta tasca s'ha dut a terme durant tot el desenvolupament del projecte).

3.2 Tecnologies

Per a dur a terme aquest projecte, principalment s'ha utilitzat el llenguatge de programació Python. També s'han utilitzat altres tecnologies per a poder completar els diferents mòduls del desenvolupament, les tecnologies utilitzades són les següents:

- Python [2]: És un llenguatge de programació d'alt nivell i actualment molt utilitzat per a temes relacionats amb el *"Big Data"* i *"Machine Learning"*. Aquest ha estat el més utilitzat durant tot el desenvolupament del projecte. Des del punt de l'extracció de la informació de Twitter, fins a l'algorítmica dels algoritmes de processament i classificació de text. Per a fer servir aquest llenguatge s'ha utilitzat l'entorn de desenvolupament Spyder [3], ja que integra moltes llibreries pel desenvolupament amb Python, de les quals moltes d'elles són útils per la recerca científica.
- API Twitter [4]: És la documentació de com connectar-se a Twitter mitjançant un llenguatge de programació. A partir d'aquesta s'han treballat dos mètodes per a l'extracció d'informació.
- MySQL: Base de dades relacional on s'ha guardat part de la informació recollida.
- MongoDB [5]: Base de dades NoSQL, utilitzades per a emmagatzemar gran quantitat de dades i per a sistemes de *"Big Data"*.
- Llibreria NLTK [6]: Es tracta d'una llibreria que ens ha servit diferents eines per a poder realitzar anàlisi i manipulació del llenguatge natural. És una llibreria destinada a l'investigació i l'ensenyament del processament del llenguatge natural.
- API Google Prediction [7]: Es tracta d'una eina creada per Google que es basa en el processament automàtic de l'anàlisi de dades. S'ha utilitzat per determinar la tendència de la informació recollida a Twitter.

4 PLANIFICACIÓ

Abans d'endinsar-nos en el desenvolupament i la implementació, es veuran quines són i com s'han planificat i estructurat les diferents fases que han fet falta pel desenvolupament del projecte. Com podem observar a la figura 2, s'ha dividit en quatre grans fases, tres d'elles són fases separades, en canvi, una d'elles ha estat necessària que fos present durant tot el procés per arribar a un anàlisi de resultats.

5 DESENVOLUPAMENT

Com s'ha pogut observar a la figura 2, per a poder desenvolupar el projecte s'han seguit un seguit de passos i processos que s'explicaran a continuació, aquests passos van des de l'extracció d'informació fins a l'anàlisi final d'aquesta, passant per processos intermedis per assegurar el seu bon funcionament. S'ha diferenciat el desenvolupament en les 4 fases existents, la part de l'ETL [8] (Extracció, transformació i emmagatzematge de dades) que inclou la fase 1, 2 i 3. La part dels algoritmes de processament de text utilitzats es troba a la fase 4 (on també s'hi inclou la fase 2).

5.1 FASE 1

Mètode d'extracció utilitzat

La primera fase que s'ha desenvolupat per a començar a treballar amb el projecte és l'extracció d'informació de la xarxa social Twitter. Per tal d'aconseguir recollir aquestes dades s'han vist els dos mètodes que ens proporciona l'API de Twitter; el mètode REST i el mètode STREAMING. Aquests dos mètodes són els més coneguts i utilitzats per a l'extracció de dades de Twitter. Cada mètode té els seus avantatges i inconvenients depenent de l'ús que se'n vulgui fer o del projecte que es porti a terme.

- El mètode REST[9]: Aquest mètode es tracta de fer una petició a l'API de Twitter segons un filtratge de *"hashtags"* o paraules clau i segons un interval de temps. Ens permet realitzar totes les accions a les quals tenim accés des de la pàgina web o aplicació creada, ens proporciona informació ja existent a Twitter a l'hora de fer-ne la petició (realitzem les peticions

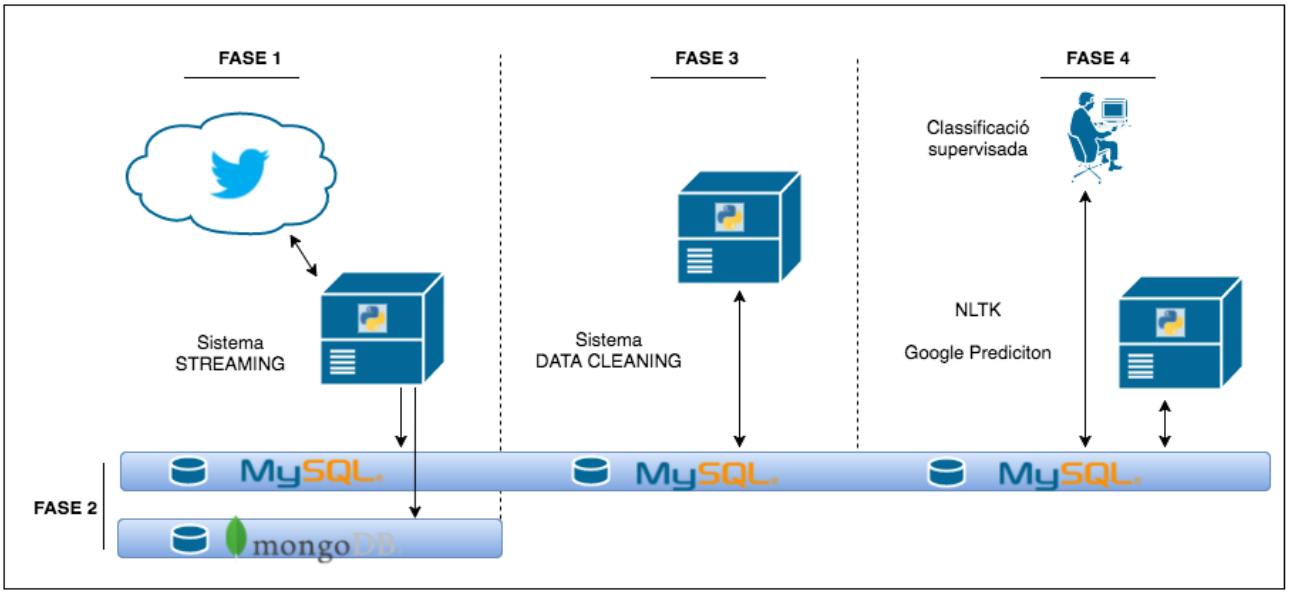

Fig. 2: Planificació per fases pel desenvolupament

mitjançant GET i POST). Un cop realitzada aquesta petició, Twitter retorna un conjunt de *tweets* que respecten els filtres introduïts prèviament. Aquest mètode es faria servir per aconseguir tots els *tweets* d'un esdeveniment ja passat, com per exemple el de les eleccions del passat 20 de desembre de 2015.

- El mètode STREAMING[10]: Aquest mètode fa l'extracció dels *tweets* a temps real. És a dir, es crea una connexió directe entre Twitter (mitjançant la seva API) i el nostre servidor. Aquesta connexió quedarà oberta a llarg termini, de forma que l'API va enviant dades i el nostre servidor les va rebent, sempre i quan compleixin el filtratge inicial, cal esmentar que no es reben el 100% dels *tweets*, ja que seria impossible segons els filtres aplicats. Aquest sistema treballa establint una connexió HTTP permanent, on la velocitat de recepció depèn de l'amplada de banda de les connexions dels dos nodes i del sobre carregament dels servidors de Twitter. En resum, amb aquest mètode es van rebent a temps real els *tweets* realitzats pels diferents usuaris segons el filtratge decidit.

Sistema STREAMING

S'ha decidit utilitzar aquest mètode en el projecte per veure que a temps real pots estar emmagatzemant dades, i que no cal esperar cert temps per a fer una petició d'un conjunt de *tweets*. També es volia observar la quantitat de *tweets* que es poden arribar a crear en tan sols unes hores fent servir aquest mètode. El tema o esdeveniment escollit per a fer l'extracció de dades va ser el passat partit de futbol Barça - Madrid el dissabte 2 d'abril del 2016.

Per a poder implementar el mètode STREAMING ha estat necessari tenir un compte a la xarxa social Twitter i registrar-se com a desenvolupador. Un cop realitzat aquest registre ha estat necessari crear una aplicació que serà on tindrem allotjat el nostre servei. Seguidament s'han de generar un seguit de codis que seran necessaris per a l'autentificació d'usuari des del mòdul d'extracció de dades (API Key, API Secret, Acces token, Acces token secret). Un cop s'han tingut aquests codis s'ha procedit a implementar el sistema Streaming amb Python. Per accedir a l'API Streaming s'ha utilitzat la llibreria ja existent de Python anomenada Tweepy[11]. És una llibreria senzilla i molt potent que ens ha permès crear aquest sistema mitjançant les seves funcions ja implementades. Tal com el procés Streaming indica, primer de tot, el nostre servidor envia una petició a Twitter fent servir els codis d'autentificació. Twitter, accepta la nostra petició i s'estableix una connexió. A partir d'aquí, comença el flux de dades entre Twitter i el nostre servidor. A mesura que es van rebent dades, aquestes són filtrades, de manera que només es processen els *tweets* segons el filtratge aplicat per l'usuari, és a dir, el nostre servidor. Un cop la informació ha estat processada, es guarda a la base de dades, en veurem el desenvolupament a la següent fase. Finalment, un cop acabada la recol·lecció, es tanca la connexió amb Twitter. Després de realitzar les proves necessàries pel bon funcionament del mòdul, aquest procés ha estat engegat i ha treballat durant aproximadament 4 hores.

Durant aquest interval de temps s'han recollit tants com 37.000 *tweets*, amb les diferents paraules decidides que tenien relació amb l'esdeveniment. El filtre de paraules utilitzades ha estat el següent: *"barça, fcb, realmadrid, clasico, elclasico, campnou, classic, gol, messi, neymar, suarez, pique, ronaldo, cr7, cruyff, benzema"*.

L'API de Twitter ens retorna la informació d'un *tweet* mitjançant una estructura de dades JSON, XML, HTML, etc. En aquest projecte s'ha decidit fer servir JSON[12]. JSON (*JavaScript Object Notation*) és un format per l'intercanvi de dades, és a dir, descriu i estructura les dades mitjançant una sintaxi dedicada que es fa servir per identificar i gestionar les dades. En cada *tweet* no només es rep el text escrit per l'usuari, sinó que cada *tweet* rebut és una estructura de dades amb molta informació (usuari, zona geogràfica, mencions, enllaços, etc.) A la figura 3 es pot veure un format reduït de la cadena rebuda amb format JSON. A l'apèndix A1, figura 10 es pot veure un *tweet* amb JSON complet.

```
"created_at":"Tue Mar 29 16:37:27 +0000 2016",
"id":714853993198067712,
"id_str":"714853993198067712",
"text":"crispuar tfg_times",
"source":"\u003ca href=\"http:\/\/itunes.apple.com\
    /us\/app\/twitter\/id409789998?mt=12\" rel=
    \"nofollow\"\u003eTwitter for Mac\u003c\/a\u003e",
"truncated":false,
"in_reply_to_status_id":null,
"in_reply_to_status_id_str":null,
"in_reply_to_user_id":null,
"in_reply_to_user_id_str":null,
"in_reply_to_screen_name":null,
```


Fig. 3: Exemple d'estructura JSON retornada per la API de Twitter

5.2 FASE 2

Emmagatzematge de dades

Seguin amb la fase 2 del projecte, tal com es pot observar a la figura 2, aquesta fase és present des de l'inici fins al final. Això és degut al fet que en les tres fases separades del projecte es necessitava l'existència d'una base de dades per a poder processar tota la informació.

Per tal de poder emmagatzemar tots els *tweets* que ens retorna Twitter mitjançant el mètode STREAMING, ens ha calgut implementar una base de dades. Aquest procés s'ha dividit en dues parts, ja que s'ha decidit fer servir dos tipus de base de dades, una base de dades estructurada i una base de dades no estructurada. Aquest fet s'ha donat per la gran utilització de les base de dades no estructurades que es fan servir per a la pràctica de *BigData*, també anomenades bases de dades NoSQL. Per tant la divisió de les dades ha estat la següent:

- Per una part tenim tots els *tweets* tal com els rebem de l'API de Twitter (en format JSON) guardats en una base de dades NoSQL com és MongoDB. MongoDB és de tipus clau/valor i document, és a dir, tenim una clau per a cada *tweet*, i el *tweet* és el valor representat en format JSON on la seva estructura no té importància per la base de dades. Això s'ha fet d'aquesta forma pel fet que ens interessa guardar tota la informació del *tweet* per a possibles consultes futures. També, donada la gran quantitat de dades que es podria arribar a emmagatzemar i la possibilitat del canvi en l'estructura de les dades que ens retorna Twitter, una base de dades NoSQL sembla la més adequada. Un altre fet característic de les bases de dades NoSQL és que actualment s'utilitzen molt en aquest tipus de projecte, especialment en sistemes *"Big Data"* i *"Machine Learning"*. Aquest tipus de base de dades permeten un disseny simple o nul, sense tenir una estructura predefinida, és a dir, les dades ja venen amb una estructura donada i la base de dades s'adapta a aquestes. També s'ha de dir que aquestes bases de dades tenen una gran escalabilitat horitzontal i un major control de la disponibilitat. Aquesta base de dades només s'ha implementat i utilitzat durant la primera i la segona fase del projecte. Ja que en les següents fases, pel tractament i processament de les dades s'ha utilitzat una SQL normal. Això s'ha fet així perquè l'objectiu del projecte no és la utilització de les bases de dades NoSQL, però com que actualment aquestes s'utilitzen molt, s'ha volgut fer una investigació de com s'utilitzaven i se n'ha realitzat una prova amb les dades extretes de Twitter.

- Per una altra part, com a base de totes les fases del projecte, s'ha fet servir una base de dades estructurada, fent servir MySQL, ja que per a la quantitat de dades utilitzades s'ajusta molt bé a les necessitats que tenim. Aquest emmagatzematge no s'ha realitzat seguint l'estructura del JSON rebut per l'API de Twitter, sinó que només s'han guardat els camps que s'han cregut necessaris per aconseguir els objectius del projecte i poder treballar millor les dades, així mantenint una estructura. La taula on hem guardat totes les dades necessàries es pot observar a la figura 4.

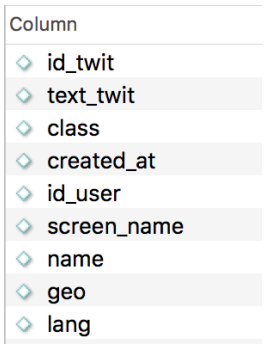

Fig. 4: Taula amb les dades necessàries del Tweet

Si en algun moment es volgués recuperar el *tweet* original per a futures implementacions o millores, podrem anar a la base de dades MongoDB a recollir-ho, ja que com s'ha dit anteriorment, en aquesta base de dades s'han guardat tots els *tweets* en format JSON. D'aquesta manera es tindrà tota la informació guardada.

5.3 FASE 3

Mòdul de Data Cleaning del *Tweet*

Per a poder realitzar un bon anàlisi de les dades recollides, ha estat necessari realitzar una normalització dels *tweets*, és a dir, fer una neteja de totes les cadenes de text que s'utilitzaran posteriorment en els algoritmes de processament i classificació d'elles, ja que sense aquesta formatació, l'anàlisi final no seria tan acotat. A part d'aplicar aquests filtres, el que també s'ha tingut en compte ha estat el llenguatge del *tweet*, ja que a la base de dades s'ha guardat un camp on s'indica l'idioma del *tweet*. Es pot dir que un anàlisi de dades amb diferents idiomes és molt difícil, i actualment sempre es fan segons un idioma seleccionat.

La formatació s'ha realitzat de la següent forma:

"RT @canchallena: Emotivo homenaje a Johan Cruyff en el Camp Nou antes del clásico español #Barça https://t.co/DAOXJa1dgq https://t.co/ebOs5cO7pf"

Aquest és un *tweet* escollit aleatòriament, com es pot veure, hi ha caràcters i paraules que podrien influir en els resul-

tats dels algoritmes de classificació de llenguatge natural. Primer de tot, s'ha fet un reemplaçament dels caràcters especials propis dels llenguatges llatins. En el cas d'aquest *tweet*, un cop passat pel nostre algoritme de reemplaçament ens quedaria de la següent manera:

"RT @canchallena: Emotivo homenaje a Johan Cruyff en el Camp Nou antes del clasico espanol #barca https://t.co/DAOXJa1dgq https://t.co/ebOs5cO7pf"

Deixem les paraules amb accents, lletres "ñï "ç"netes, amb les lletres de l'alfabet anglosaxó.

Un cop passat aquest filtre, s'ha observat que els enllaços a pàgines o imatges no formen part del llenguatge natural, per tant s'ha decidit treure'ls. Aquest procés s'ha realitzat mitjançant les expressions regulars, de tal forma que el *tweet* queda de la següent manera:

"RT @canchallena: Emotivo homenaje a Johan Cruyff en el Camp Nou antes del clasico espanol #barca"

Encara no tenim el *tweet* formatat apropiadament, a part d'excloure'n els enllaços també s'ha decidit eliminar les mencions dels usuaris. En aquest punt es podrien analitzar els *tweets* de l'usuari mencionat però no és l'objectiu del treball.

"RT: Emotivo homenaje a Johan Cruyff en el Camp Nou antes del clasico espanol #barca"

Per acabar de formatar el *tweet*, s'ha decidit treure també els signes de puntuació i el conjunt de lletres "RT", quedant de la següent manera.

"Emotivo homenaje a Johan Cruyff en el Camp Nou antes del clasico espanol #barca"

Un cop s'han tingut tots els *tweets* formatats de la forma vista anteriorment, s'han guardat de nou en la base de dades MySQL, actualitzant el camp del text anterior. A partir d'aquest punt, les operacions que es fan amb els *tweets* ja dependrà del tipus de classificador utilitzat. Hi ha algoritmes que ja porten funcions i eines per a fer aquesta tasca.

5.4 FASE 4

Classificació manual de *Tweets*

Per tal de poder obtenir uns resultats de totes les dades recollides, en aquest projecte s'ha decidit posar en pràctica l'aprenentatge supervisat. És a dir, ha estat necessari tenir una col·lecció de *tweets* classificats, per tant, de totes les dades recollides s'ha realitzat una prèvia classificació supervisada seguint un criteri personal, on ha influït la subjectivitat. S'han definit tres categories per a la classificació: *tweet* negatiu, *tweet* neutre i *tweet* positiu. Aquesta classificació es dóna per vàlida i és la informació en què es basarà l'anàlisi dels *tweets* no classificats. Per una banda es proporciona un conjunt de dades classificades i per l'altre, es proporciona un conjunt de dades sense classificar, que gràcies als algoritmes de classificació de text i llenguatge natural, mitjançant un entrenament amb el conjunt de *tweets* classificats ens donarà un resultat de classificació. Ja existeixen exemples i conjunts de dades classificades per a realitzar aquest tipus d'activitat, però dins l'àmbit d'aquest projecte s'ha volgut fer la classificació segons el tema escollit.

Algoritmes de processament de text utilitzats (NLTK i API Google Prediction)

Hi ha molts algoritmes i programes per a realitzar tasques de *"Big Data"* i *"Machine Learning"*, ja que actualment moltes empreses creen o modifiquen algoritmes ja existents. Per a poder observar la tendència i veure resultats de totes les dades que s'han recollit i formatat, s'ha decidit utilitzat dos algoritmes o API's, aquests dos són els següents:

- NLTK: Primerament s'ha utilitzat la llibreria NLTK, que incorpora un conjunt d'eines pel tractament i l'anàlisi del llenguatge natural.

  Per a poder fer servir el classificador d'aquesta llibreria cal generar una estructura de dades determinada per la llibreria a partir dels *tweets* classificats i normalitzats. Per poder crear aquesta estructura cal aplicar un preprocés a cada *tweet*.

  La mateixa llibreria ens ofereix les funcions necessàries per dur a terme aquestes tasques. Primerament apliquem un algoritme de *"tokenization"* que consisteix a separar el *tweet* en paraules individuals a partir d'espais en blanc, símbols, etc. Aquest procés retorna les paraules del *tweet* en un vector. Després es passa aquest vector per un procés d'*"StopWords"* que consisteix a eliminar totes les paraules sense significat com articles, pronoms, preposicions, etc. La llibreria ens aporta una eina amb diccionaris predefinits que ens eliminen aquest tipus de paraules segons l'idioma tractat. Per acabar de tenir el *tweet* tal com és necessari, cal afegir la categoria a cada vector, en neutre, positiu o negatiu.

  Tots aquests processos es fan mitjançant una classe. Fent-la servir, es genera un conjunt de dades per cada categoria de classificació: positiu, negatiu i neutre. Aquests conjunts de dades es recullen directament de la base de dades. Posteriorment es genera el conjunt d'entrenament agafant 3/4 de cada categoria. El 1/4 sobrant es farà servir com a conjunt de test. Seguidament s'entrena amb el conjunt d'entrenament el classificador Naive Bayes[13]. Ara, el classificador pot ser avaluat amb el conjunt de test per obtenir mètriques com la matriu de confusió o es pot fer servir per avaluar *tweets* individuals fora d'aquests conjunts, és a dir amb el conjunt de *tweets* no classificats

- API Google Prediction

  Com a alternativa, s'ha escollit l'API de Google Prediction. S'ha decidit utilitzar aquesta eina perquè amb ella es pot fer pràcticament qualsevol cosa relacionada amb l'anàlisi de dades. Des d'un sistema de recomanació, fins a un anàlisi de sentiments, passant per la detecció d'Spam. Aquesta és una API de pagament, però és possible utilitzar-la durant 60 dies de manera gratuïta; una bona ocasió per a investigar a fons l'eina. Per a poder-la fer servir és necessari tenir un compte actiu de Google i crear un projecte al *"Google Cloud Platform"* i tenir activats els dos mòduls de *"Google Prediction API"* i *"Google Cloud Storage*

API". Aquesta API permet fer servir les seves eines a partir d'un panell de gestió *online*. També hi ha la possibilitat fer servir aquesta API implementant un mòdul amb Python, aquest es veurà més endavant.

Pel funcionament d'aquesta API és necessari crear un projecte, on a partir d'aquest, es treballaran les dades introduïdes, mitjançant l'entrenament i la validació i anàlisi dels resultats. Per poder utilitzar l'eina es farà servir el conjunt de *tweets* que prèviament s'han classificat, aquests caldrà introduir-los al sistema mitjançant un format CSV, ja que és com els accepta l'API. A continuació hi ha un seguit de passos per a poder entrenar i preveure resultats l'algoritme.

- insert: en aquest primer mòdul s'envia tota la informació del CSV, així guardant-la i entrenant l'algoritme. Podem observar com és la crida a la figura 5

```
POST
https://www.googleapis.com/prediction/v1.6/projects/testtwitter-1312/trained
models?key={YOUR_API_KEY}
{
"id": "test_twits",
"storageDataLocation": "nahui/twits_class.txt"
}
```

Fig. 5

- get: aquí es podrà veure si l'entrenament ha finalitzat, si és així, es retorna una resposta amb els resultats de la tasca. Ho podem observar a la figura 6

```
{
"kind": "prediction#training",
"id": "test_twits",
"selfLink":
"https://www.googleapis.com/prediction/v1.6/projects/testtwitter-1312/trained
models/test_twits",
"created": "2016-05-21T15:37:38.905Z",
"trainingComplete": "2016-05-21T15:38:17.363Z",
"modelInfo": {
"numberInstances": "853",
"modelType": "classification",
"numberLabels": "3",
"classificationAccuracy": "0.64"
},
"trainingStatus": "DONE"
}
```

Fig. 6

Es pot observar que l'entrenament ha estat amb un conjunt de 853 *tweets*, utilitzant un model de classificació i, classificant-los en tres categories (neutre, positiu i negatiu). També es pot observar que la precisió del classificador és d'un 64%, per tant es pot dir que classifica correctament 64% de *tweets* del conjunt de test.

- predict: en aquest punt s'envia el *tweet* per tal que l'algoritme l'etiqueti en neutre, positiu o negatiu. També ens retorna el percentatge de cada categoria. Com podem observar a la figura 7, ha etiquetat el *tweet* enviat en neutre.

Un cop vist el procediment per a extreure resultats des de la consola que ens facilita l'API de Google Prediction, s'ha decidit implementar-la amb Python, aquesta

```
{
"kind": "prediction#output",
"id": "test_twits",
"selfLink":
"https://www.googleapis.com/prediction/v1.6/projects/testtwitter-1312/trained
models/test_twits/predict",
"outputLabel": "neutro",
"outputMulti": [
    {
    "label": "neutro",
    "score": "0.666667"
    },
    {
    "label": "positivo",
    "score": "0.333333"
    },
    {
    "label": "negativo",
    "score": "0.000000"
    }
    ]
}
```

Fig. 7

tasca s'ha fet per a poder classificar tot el conjunt de *tweets* que tenim sense una classificació supervisada. Per a poder implementar i fer servir l'API de Google Prediction, ha estat necessari crear uns codis d'autentificació. Amb aquests codis podrem fer les peticions des d'un entorn extern a la consola de Google.

6 RESULTATS

Com a resultat principal del projecte, s'ha fet una classificació de *tweets* basada en els sentiments de les persones davant un important esdeveniment esportiu.

S'ha definit un protocol per a l'extracció, transformació i càrrega (ETL) de les dades recollides. L'extracció de les dades s'ha dut a terme mitjançant un mòdul a temps real en Streaming, que pel seu posterior anàlisi s'ha emmagatzemat a una base de dades relacional MySQL. A part, també s'ha implementat una base de dades NoSQL per a possibles línies futures del projecte.

Amb l'objectiu d'obtenir el millor resultat possible i no veure's influenciat per elements del llenguatge no natural, s'han eliminat paraules referenciades només utilitzades a la xarxa social Twitter. A més a més, s'han eliminat paraules sense un significat propi per tenir un resultat més real.

S'ha creat un conjunt d'entrenament i test segons una classificació supervisada per un humà, avaluant el sentiment de cada *tweet* de la manera més objectiva possible.

Un altre resultat és que s'han implementat dos algoritmes de classificació i processament de text extraient la seva precisió per a la posterior classificació dels *tweets*.

Per una banda s'ha implementat la llibreria NLTK per a classificar el conjunt de *tweets* no supervisats. Utilitzant els seus mètodes pel processament del llenguatge natural i classificant els *tweets* mitjançant el seu classificador Bayesià en un entorn local.

Per altra banda, s'ha fet servir el classificador de l'API de

Google Prediction per classificar el conjunt de *tweets*. Per fer això s'ha utilitzat la consola de l'API de Google Prediction per fer les peticions al sistema. Una vegada vist aquest funcionament s'ha implementat un script local amb Pyhton per a poder classificar tots els *tweets* no supervisats.

Exemples de classificació

A continuació, es mostraran els resultats de la classificació supervisada. S'ha realitzat aquesta classificació amb un total de 853 *tweets*. Tal com podem observar a la figura 8, els resultats no són gaire equilibrats, ja que podem veure una gran tendència a *tweets* neutres i positius, envers els negatius. S'ha escollit aquest conjunt de dades per plasmar els sentiments reals davant d'un esdeveniment esportiu.

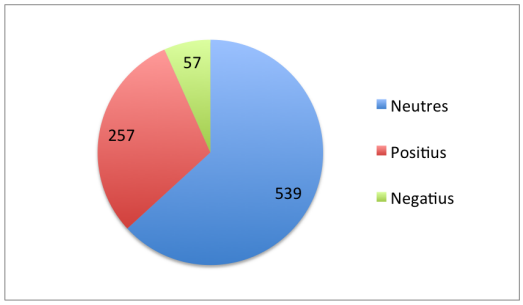

Fig. 8: Classificació supervisada

Per altra banda, es mostraran els resultats de la classificació d'un conjunt d'uns 8000 *tweets* sense categoritzar. Es faran servir els classificadors NLTK i API Google Prediction, tots dos han estat entrenats amb el mateix conjunt de *tweets* etiquetats.

Els resultats del conjunt de test els podem observar a la taula 1 per l'algoritme NLTK, i a la taula 2 per l'API de Google Prediction.

	Negatius	Positius	Neutres
Negatius	81%	28%	33%
Positius	12%	54%	22%
Neutres	17%	16%	45%

Taula 1: Matriu de confusió del l'algoritme NLTK

	Negatius	Positius	Neutres
Negatius	18%	0%	3%
Positius	21%	48%	16%
Neutres	61%	52%	81%

Taula 2: Matriu de confusió de l'API de Google Prediction

Per finalitzar la visualització dels resultats, podem observar la figura 9, on hi ha la classificació dels *tweets* no etiquetats pels dos algoritmes implementats. Es pot apreciar que amb l'NLTK els resultats són més equilibrats i homogenis, en canvi, amb l'API de Google Prediction, els resultats són semblants a la classificació no supervisada. No es pot saber exactament la diferència d'aquest fet, ja que l'API de Google és de codi propietari i no podem veure com treballar internament. No se sap com treballen internament els algoritmes.

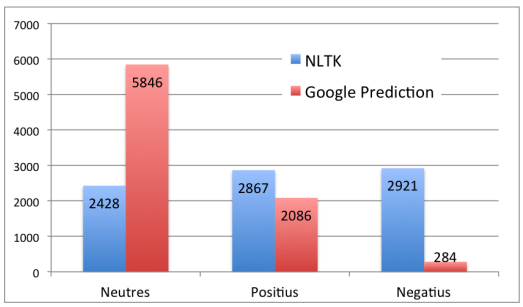

Fig. 9: Classificació mitjançant algoritmes de classificació

7 CONCLUSIONS

Desenvolupar aquest TFG ha suposat investigar molt sobre les tecnologies utilitzades, ja que en un principi, part d'elles, com han estat les API's, es desconeixien. En canvi, utilitzar el llenguatge de programació Python, no ha suposat d'investigació, ja que es tenia una mínima experiència.

Per començar, en el mòdul d'extracció de dades de Twitter a temps real, s'ha observat que existeix un rendiment alt, ja que en el moment en què un usuari realitzava un comentari sobre el tema escollit, l'API retornava resposta instantàniament. Gràcies a aquest objectiu, s'ha après a utilitzar l'API de Twitter juntament amb les eines que té.

Cal destacar la classificació supervisada que s'ha dut a terme d'un conjunt de *tweets*, ja que gràcies a això s'ha observat la tendència de la mostra recollida segons un criteri humà. Després de realitzar aquesta tasca, s'ha arribat a la conclusió que els comentaris de les persones són més positius i neutres, que no pas negatius. Degut ha aquest fet, els algoritmes de processament de text i llenguatge natural tindran la tendència a donar resultats positius i neutres, ja que s'ha treballat amb una mostra on existeixen pocs *tweets* negatius.

També és important destacar la feina feta en la normalització dels *tweets*, ja que inicialment no es va tenir en compte. Cosa que ha comportat la investigació de com fer-ho. Realment aquesta ha estat una de les tasques més importants, ja que, els algoritmes de processament de text no tenen en compte els enllaços, els signes de puntuació, les mencions d'usuaris, etc. És a dir, existeix molta informació sense significat a l'hora de realitzar aquests tipus d'anàlisi.

Aplicant els dos algoritmes i fent diferents tipus de proves, s'ha observat que els resultats són més positius i neutres, és a dir, les opinions i sentiments de les persones a la xarxa social Twitter davant un esdeveniment esportiu son més de felicitat que no de tristesa.

Com a línies futures del projecte, es podria fer l'anàlisi de les imatges adjuntades a un *tweet*, així com la relació entre els usuaris que estan comentant sobre el mateix tema.

AGRAÏMENTS

Els agraïments d'aquest projecte són principalment pel meu tutor Jordi Casas Roma, perquè gràcies als seus consells s'-han pogut realitzar els objectius bàsics proposats, així aconseguint un bon resultat. També vull donar les gràcies a la família i als amics per tot el suport rebut durant aquests anys a la universitat i durant el TFG.

REFERÈNCIES

[1] Anàlisi de popularitat dels resultats electorals, per a WEBSAYS. https://websays.com/es/politica/el-analisis-big-data-de-la-conversacion-digital-como-herramienta-para-comprender-procesos-electorales/

[2] Documentació Python 2.X. https://docs.python.org/2/

[3] Documentació Spyder https://pythonhosted.org/spyder/

[4] Documentació i API's Twitter https://dev.twitter.com/overview/documentation

[5] Que és i com funciona MongoDB http://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no

[6] Documentació NLTK http://www.nltk.org/

[7] Documentació API Google Prediction https://cloud.google.com/prediction/docs/apis#v16

[8] ETL, Extract, Transform and Load https://es.wikipedia.org/wiki/Extract,_transform_and_load

[9] Documentació API REST Twitter https://dev.twitter.com/rest/public

[10] Documentació API STREAMING Twitter https://dev.twitter.com/streaming/overview

[11] Documentació llibreria Tweepy http://docs.tweepy.org/en/v3.5.0/

[12] Introducció a JSON http://www.json.org/

[13] Naive Bayes Classifier https://en.wikipedia.org/wiki/Naive_Bayes_classifier

APÈNDIX

A.1 JSON Complet


```
{
   "created_at":"Tue Mar 29 16:37:27 +0000 2016",
   "id":714853993198067712,
   "id_str":"714853993198067712",
   "text":"crispuar tfg_times",
   "source":"\u003ca href=\"http:\/\/itunes.apple.com\
        /us\/app\/twitter\/id409789998?mt=12\" rel=
        \"nofollow\"\u003eTwitter for Mac\u003c\/a\u003e",
   "truncated":false,
   "in_reply_to_status_id":null,
   "in_reply_to_status_id_str":null,
   "in_reply_to_user_id":null,
   "in_reply_to_user_id_str":null,
   "in_reply_to_screen_name":null,
   "user":{
      "id":273051441,
      "id_str":"273051441",
      "name":"Cristian Puig",
      "screen_name":"crispuar",
      "location":null,
      "url":null,
      "description":"Catalunya & #puigtureig",
      "protected":false,
      "verified":false,
      "followers_count":125,
      "friends_count":420,
      "listed_count":4,
      "favourites_count":121,
      "statuses_count":198,
      "created_at":"Sun Mar 27 18:39:00 +0000 2011",
      "utc_offset":10800,
      "time_zone":"Athens",
      "geo_enabled":false,
      "lang":"ca",
      "contributors_enabled":false,
      "is_translator":false,
      "profile_background_color":"022330",
      "profile_background_image_url":"http:\/\/abs.twimg.com\/images\/themes\/theme15\/bg.png",
      "profile_background_image_url_https":"https:\/\/abs.twimg.com\/images\/themes\/theme15\/bg.png",
      "profile_background_tile":false,
      "profile_link_color":"FA4949",
      "profile_sidebar_border_color":"A8C7F7",
      "profile_sidebar_fill_color":"C0DFEC",
      "profile_text_color":"074707",
      "profile_use_background_image":true,
      "profile_image_url":"http:\/\/pbs.twimg.com\/profile_images\/478607985485574144\/r-ljkc90_normal.jpeg",
      "profile_image_url_https":"https:\/\/pbs.twimg.com\/profile_images\/478607985485574144\/r-
          ljkc90_normal.jpeg",
      "default_profile":false,
      "default_profile_image":false,
      "following":null,
      "follow_request_sent":null,
      "notifications":null
   },
   "geo":null,
   "coordinates":null,
   "place":null,
   "contributors":null,
   "is_quote_status":false,
   "retweet_count":0,
   "favorite_count":0,
   "entities":{
      "hashtags":[

      ],
      "urls":[

      ],
      "user_mentions":[

      ],
      "symbols":[

      ]
   },
   "favorited":false,
   "retweeted":false,
   "filter_level":"low",
   "lang":"cy",
   "timestamp_ms":"1459269447980"
}
```


Fig. 10: Estructura d'un Tweet en JSON complet