Anàlisis de sentiments a Twitter en l'àmbit cinematogràfic

Roger Gimeno Pagès

Resum— Amb el pas dels anys i l'evolució de les tecnologies, el sector del cinema li han sorgit diferents problemes que fan que en aquest sector no es puguin cometre grans errors. La pirateria i els elevats impostos en el sector de la cultura són uns dels grans problemes que te el cinema actualment. L'evolució de les tecnologies també ha comportat grans avanços en l'anàlisi de dades on es pot obtenir grans quantitats d'informació. Avui en dia, la informació s'ha tornat un recurs molt valuós per a les empreses a la hora de crear estratègies de futur. En el nostre projecte crearem informació útil per a empreses del àmbit cinematogràfic, on agafarem l'opinió que te la gent sobre una pel·lícula que es vagi a estrenar o ja s'hagi estrenat, i l'analitzarem per poder saber quina és l'opinió que tenen els espectadors sobre aquesta pel·lícula. Per a poder analitzar l'opinió de la gent, analitzarem els sentiments del tweets dels espectadors que utilitzin el *hashtag* de la pel·lícula en qüestió, podent així obtenir l'opinió dels espectadors sobre una pel·lícula.

Paraules clau— Anàlisis de seniments, anàlisis d'opinions, MongoDB, Twitter, Tweepy, tweet, NLTK, Business Intelligence, reporting, cine, pel·lícula, hashtag.

Abstract— Through the years and the evolution of technology , the film industry have been several problems that make this industry can not make big mistakes. Piracy and high taxes in the culture sector is a major problem that has cinema today. The evolution of technology has also led to great advances in the analysis of data where you can get large amounts of information. Today, information has become a valuable resource for companies in creating future strategies. In our project we will create useful information for companies in the film industry where we take the view that people have about a film which is to be released or have already been released and analyze it to know the movie viewers opinion about this film. In order to analyze the opinion of the people, we will analyze the feelings of the viewers tweets that uses the hashtag of the film in question, then we can get the opinion of spectators on a film.

dex Terms— Seniment analysis, opinion analysis, MongoDB, Twitter, Tweepy, tweet, NLTK, Business Intelligence, reporting, film, hashtag.

1 Introducció

AVUI en dia, molta gent comenta les seves experiències o pensaments sobre una pel·lícula mitjançant les xarxes socials.

Tots aquests comentaris que fa la gent sobre una pel·lícula en concret és informació molt valuosa que es podria utilitzar per saber si una pel·lícula ha agradat o no al públic, que pensa el públic d'una edat concreta, i moltes altres preguntes que es podrien respondre analitzant aquesta informació.

La xarxa social que utilitzarem en el projecte és Twitter, que es tracta d'una xarxa on els usuaris escriuen petits missatges anomenats *tweets* que són vistos per els usuaris que els segueixen. Una de les característiques principals que te Twitter és que els seus usuaris expliquen les seves vivencies i opinions utilitzant un *hashtag*. On un *hashtag* és una etiqueta que representa un tema en concret.

Per això, hem vist l'oportunitat de fer aquest projecte on analitzarem tots aquests comentaris que fa la gent, per poder tenir molta més informació i així poder respondre les preguntes que ens sorgeixin sobre el que pensa el públic, per després (en cas de que el nostre projecte ho utilitzes una productora) poder fer estratègies de futur segons els resultats obtinguts.

Aleshores, el projecte es basarà en obtenir tots aquells *tweets* on es faci un comentari sobre alguna pel·lícula, emmagatzemar-los, analitzar-los i classificar-los segons si el contingut és o no favorable, i mostrar aquesta informació d'una forma comprensible per a qualsevol persona. Al finalitzar aquests processos, haurem transformat les dades que hem recol·lectat, emmagatzemat i processat en informació útil.

Al llarg d'aquest document mostrarem l'abast del projecte, l'estat de l'art, la metodologia utilitzada, el desenvolupament, els resultats obtinguts i una conclusió del projecte on parlarem de possibles extensions que aquest podrà tenir.

- *E-mail de contacte: rogergimenop@gmail.com*

- *Menció realitzada: Tecnologies de la Informació.*
- *Treball tutoritzat per: Jordi Casas Roma (Tecnologies de la Informació)*
- *Curs 2015/16*

2 ABAST DEL PROJECTE

2.1 Objectiu general

L'objectiu principal del nostre projecte és aconseguir, analitzar i mostrar informació valuosa mirant els comentaris que fa la gent a Twitter sobre una pel·lícula en concret, així grans empreses com ara productores, sales de cine o algun particular, podrà utilitzar aquesta informació que li brindem per a poder fer diferents estratègies segons les critiques de la gent sobre una pel·lícula.

2.2 Objectius específics

Per poder assolir l'objectiu principal, hem fet un llistat de objectius específics i quina és la importància que tenen dins del projecte:

- Obtenir i filtrar *tweets* segons un *hashtag* concret. (O1)
- Emmagatzemar els *tweets* dels usuaris de Twitter. (O2)
- Emmagatzemar les dades obtingudes en una Base de dades (BBDD), considerant la possibilitat d'utilitzar un BBDD NoSQL, que permetria escalar a grans volums d'informació (Big Data). (O3)
- Anàlisis de sentiments dels *tweets* amb un *hashtag* concret (classificació de comentari en favorable, desfavorable o neutre). (O4)
- Analitzar l'impacte i com minimitzar la ironia i el sarcasme. (O5)
- Mostrar els *tweets* i els resultats de l'anàlisis de sentiments amb un *hashtag* en concret. (O6)
- Incloure característiques dels *tweets* a la hora de mostrar els resultats. (O7)

2.3 Beneficis previstos

Un cop finalitzat el projecte, aquest pot produir una sèrie de beneficis a les empreses o particulars que utilitzin l'aplicació resultant del projecte. Hem fet una llista de possibles beneficis que pot oferir la nostra aplicació.

- Mostrar si una pel·lícula ha sigut ben rebuda per la gent o si no ho ha sigut.
- Mostrar quines son les expectatives de la gent sobre una pel·lícula que encara no s'ha estrenat.
- Informació per a fer estratègies de futures pel·lícules segons l'opinió de la gent sobre aquesta.
- Informació a les sales de cinema per a establir el temps que estarà disponible la pel·lícula al cinema segons l'opinió de la gent abans i després de la seva estrena.
- Mostrar el tipus de persona que li ha agradat una pel·lícula en concret.

2.4 Prioritat dels objectius

A la Taula 1, podem observar les diferents prioritats que tenen els objectius del projecte.

TAULA 1
LLISTAT DE LES PRIORITATS DELS OBJECTIUS

Objectiu	Crític	Principal	Secundari
O1	X	X	
O2		X	
O3		X	
O4	X	X	
O5			X
O6		X	
O7			X

3 ESTAT DE L'ART

En aquest apartat parlarem de diferents aplicacions que hi han al mercat que utilitzen l'anàlisi de sentiments a Twitter. Mirarem quines són les seves funcionalitats per veure si podríem incorporar alguna d'elles en el nostre projecte, és a dir, analitzarem el mercat per veure si no havíem pensat en alguna funcionalitat que ens seria útil en el nostre projecte.

3.1 Brandwatch Analyze

Començarem amb una aplicació de l'empresa Brandwatch anomenada Brandwatch Analyze [1]. Aquesta aplicació analitza les dades de les xarxes socials per a poder mesurar i optimitzar les accions de màrqueting utilitzant la plataforma de social intelligence. No analitza només Twitter, sinó que analitza també altres xarxes socials com pot ser facebook o fins i tot les noticies.

Entre les funcionalitats que ofereix i que més ens interessen, podem destacar-ne algunes: té una cobertura multi llenguatge, és a dir, pot analitzar *tweets* en diferents idiomes, ens dona la localització de les dades, té una interfície ràpida i intuïtiva, i ens deixa personalitzar els dashboards on se'ns mostra la informació. També fa el més important per nosaltres, que seria la anàlisis de sentiments, que en el seu cas, ens diu si un comentari és favorable, desfavorable o neutre.

Aquestes funcionalitats citades anteriorment, es podrien tenir en compte a la hora de plantejar el nostre projecte, ja que ens podrien ser molt beneficioses i donar un plus de qualitat al projecte.

3.2 SocialMention

Una altre aplicació que esta al mercat és l'anomenada SocialMention [2]. És una aplicació gratuïta i molt més senzilla que la vista anteriorment. En aquest cas, és un portal web on poses el que vols que sigui buscat per analitzar els comentaris sobre el que has posat, i es basa en quatre paràmetres principals: la passió, que seria la quantitat d'autors que hi parlen; l'abast, on es mostraria la influencia; el sentiment, on es mostra les mencions positives y negatives; i la força, on es mostra les frases mencionades en les ultimes 24 hores.

És un aplicació bastant senzilla, però amb unes funcionalitats diferents a les que hem vist a l'aplicació Brandwatch Analyze. Té unes funcionalitats interesants les quals podríem utilitzar en el nostre projecte.

4 Metodologia

Pel poc temps que tenim per desenvolupar el projecte i les característiques d'aquest, utilitzarem una única metodologia de treball per la seva realització i desenvolupament. Aquesta metodologia és la metodologia de desenvolupament en cascada, que es basa en ordenar les fases del projecte de tal forma que l'inici de cada una d'aquestes fases hagi d'esperar la finalització de la fase anterior.

La raó per haver escollit aquesta metodologia és perquè cada una de les fases del projecte necessita de la fase anterior per poder-la realitzar. També hauríem d'afegir, que cada una de les fases on es desenvolupa el software del projecte, es testejarà i es comprovarà que funciona correctament abans de passar a la següent fase.

El projecte esta dividit en les següents 4 fases:

4.1 Fase d'anàlisi i d'estudi (Fase 1)

En aquesta fase es definirà totes aquelles coses que poden ser rellevants pel projecte i per definir d'una forma clara com farem i que utilitzarem a les següents fases.

4.2 Fase d'obtenció, filtratge i emmagatzematge dels *tweets* (Fase 2)

En aquesta fase obtindrem els *tweets* que volem analitzar, per això els haurem de filtrar segons un hashtag en concret (el nom de la pel·lícula que volem). Un cop tinguem els *tweets* filtrats, els guardarem en una BBDD per poder accedir-hi a ells en el moment de l'anàlisi i quan mostrem els resultats.

4.3 Fase d'anàlisi de sentiments i d'opinions (Fase 3)

En aquesta fase analitzarem cada un dels *tweets* que hem emmagatzemat en la fase anterior i els classificarem segons si són favorables, desfavorables o neutres a una pel·lícula en concret. El resultat de l'anàlisi també s'emmagatzemarà en una base de dades, així podrem utilitzar els resultats de l'anàlisi sempre que volem sense haver de analitzar tots els *tweets* un altre cop.

4.4 Fase de reporting (Fase 4)

En aquesta fase mostrarem els resultats de l'anàlisi mitjançant una interfície senzilla i fàcil d'entendre, donant el màxim d'informació i característiques dels usuaris que creiem importants.

Per a fer el seguiment del projecte utilitzarem el Microsoft Project [3] per tenir una visió global dels terminis de cada una de les fases i per saber en cada moment quina tasca hem de realitzar.

4.5 Planificació

A la Taula 2, podem observar la planificació que s'ha dut a terme durant el projecte.

TAULA 2
PLANIFICACIÓ DEL PROJECTE

Tasca	Durada	Inici	Fi
Planificació	11 dies	19/02/16	05/03/16
Fase 1	10 dies	06/03/16	16/03/16
Anàlisis	4 dies	06/03/16	09/03/16
Estudi de viabilitat	3 dies	10/03/16	13/03/16
Pressupost	3 dies	14/03/16	16/03/16
Fase 2	24 dies	16/03/16	17/04/16
Obtenció i filtratge de *tweets*	8 dies	17/03/16	27/03/16
Emmagatzemar *tweets*	12 dies	28/03/16	12/04/16
Testeig	4 dies	13/04/16	17/04/16
Fase 3	11 dies	18/04/16	02/05/16
Anàlisis del *tweets*	6 dies	18/04/16	24/04/16
Classificació dels *tweets*	4 dies	25/04/16	28/04/16
Testeig	2 dies	29/04/16	02/05/16
Fase 4	16 dies	02/05/16	22/05/16
Crear *report*	10 dies	02/05/16	13/05/16
Afegir característiques dels usuaris	3 dies	14/05/16	17/05/16
Testetig	3 dies	19/05/16	22/05/16

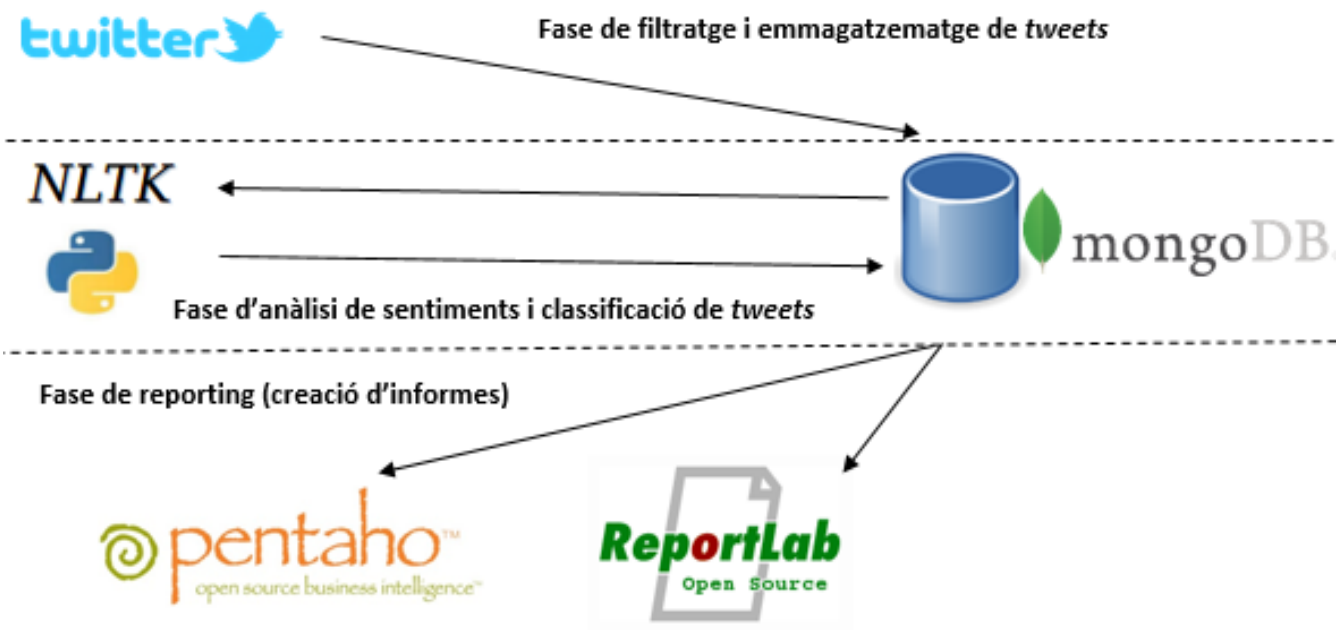

Fig.1: Fases de l'etapa de desenvolupament.

5 DESENVOLUPAMENT

Un cop acabada la fase d'anàlisi i els objectius, començarem l'etapa de desenvolupament. Aquesta etapa està formada per la fase 2, 3 i 4 del projecte. En elles assolirem els objectius del projecte proposats anteriorment.
A la Figura 1, podem veure quines són les diferents fases, algunes de les eines que utilitzarem, i quin serà el flux de dades per arribar al resultat proposat.

5.1 Recol·lecció i emmagatzematge de *tweets*

La segona fase del projecte es basa en la recol·lecció de *tweets* de la base de dades de Twitter i emmagatzemar-los en una Base de dades pròpia, a poder ser NoSQL. En aquesta fase realitzarem els tres primers objectius del projecte, que són:

- Obtenir i filtrar *tweets* segons un *hashtag* concret.
- Emmagatzemar els *tweets* dels usuaris de Twitter.
- Emmagatzemar les dades obtingudes en una BBDD, considerant la possibilitat d'utilitzar un BBDD NoSQL, que permetria escalar a grans volums d'informació (Big Data).

Per realitzar aquesta fase, l'hem dividit en dues subfases que serien les dues tasques a realitzar. En la primera subfase (recol·lecció de *tweets*) realitzaríem el primer objectiu dels tres mostrats anteriorment, i en la segona subfase (Emmagatzematge de *tweets*) realitzaríem els dos objectius restants. Aquestes fases, com tota la part de programació del projecte, està escrita amb el llenguatge de programació Python.

5.1.1 Recol·lecció de *tweets*

Per recol·lectar *tweets* de la base de dades de Twitter, hem utilitzat una llibreria de Python anomenada Tweepy.

Tweepy és una llibreria que s'utilitza per accedir a la API de Twitter [4]. Hem escollit aquesta llibreria ja que proporciona un conjunt de funcions simples i intuïtiva que és a la hora de treballar amb ella. Simplement hem de seguir uns petits passos per a poder-nos connectar a la API de Twitter gràcies a Tweepy.

Un cop instal·lada, hem de registrar-nos a Twitter i registrar també la nostre aplicació per així obtenir les claus necessàries per autenticar-nos a la API de Twitter. Un cop registrats sens donarà una clau anomenada API Key i una altre que es diu API Secret, i les utilitzarem per autenticar-nos. Ens haurem de autenticar cada cop que utilitzem la llibreria.

Després de realitzar la preparació anterior, ja ens podem connectar a la API de Twitter i fer les peticions que volem. Primer de tot hem de saber com funciona la API de Twitter, la API a la que podem accedir des de Tweepy es diu API REST [5], que ens dona accés a llegir i escriure dades de Twitter com ara *tweets*, els usuaris que han fet un *tweet* en concret i diferents dades de cada un dels *tweets*. Cada un dels *tweets* estaran disponibles en format JSON.

A part de la API REST també podem emprar la API Streaming, però no ens interessa tant com la REST, ja que ens dona *tweets* en temps real i això estaria més ambientat en projectes basats en events, com ara un partit de futbol.

Un cop ja sabem que és Tweepy i la API REST de Twitter, hem començat a recollir els primer *tweets*. Per fer-ho, hem

utilitzat una funció que ens permet filtrar *tweets* que tinguin una paraula en concret. En el nostre cas, hem utilitzat aquesta funció posant com a paraula clau el *hashtag* de la pel·lícula que volem fer-li l'anàlisi de sentiments. La funció ens va retornant *tweets* amb la paraula clau que volem, però té un problema, esta limitada a un numero de *tweets* 15 minuts, fent més difícil la tasca de recol·lecció.

Per solucionar aquest problema, hem fet un bucle on es pari de fer peticions per rebre *tweets* quan s'arribi al límit, fins que no passin els 15 minuts per poder-ne fer més. Així podem anar recol·lectant tants *tweets* com volem. Podem veure la solució en l'Apèndix 2.
Un cop arribat en aquest punt, ja podem passar a la següent subfase per procedir a emmagatzemar aquests *tweets* per després poder analitzar-los.

5.1.2 Emmagatzematge de *tweets*

En aquesta subfase hem realitzat els dos últims objectius de la fase, que serien emmagatzemar els *tweets* i que aquesta *tweets* s'emmagatzemin en una BBDD a poder ser NoSQL.
Primer de tot vam mirant quins tipus de BBDD NoSQL existeixen, per així agafar el tipus que més ens convenia [6]. Hem vist que hi ha quatre tipus: orientada a documents, orientada a columnes, clau valor i en grafs.

Per començar, vam descartar la orientada a grafs, perquè en cap moment necessitem veure les relacions que hi ha entre els *tweets*, sinó que volem guardar diferents dades de *tweets* sense que estiguin relacionats entre ells mitjançant un graf. D'aquesta mateixa forma hem descartat la del tipus clau valor, ja que només amb el camp valor no tindríem les dades suficients per analitzar els *tweets*.

Així que dels dos tipus que falten, ens vam decantar per l'orientada a documents, ja que ho podríem estructurar de la següent manera: les bases de dades orientades en documents estan basades en tenir col·leccions, on aquestes col·leccions contenen documents que serien les files d'una base de dades SQL. Aquests documents estan estructurats com si fossin diccionaris en el llenguatge Python. Nosaltres el que farem és que cada *hashtag* sigui una col·lecció, i cada document serà un diccionari amb els dades necessàries de un *tweet*, així tindrem tots els *tweets* d'un *hashtag* guardats dins d'una col·lecció que serà el propi h*ashtag*. Podem veure l'estructura en la Figura 2.

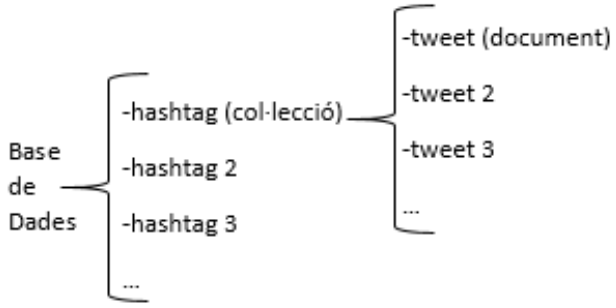

Fig.2: Estructura de la base de dades.

La base de dades que utilitzarem és MongoDB [7], perquè de les bases de dades orientades a objectes és la més madura que hi ha i la més utilitzada. Això ens dona seguretat en el sentit de que el seu funcionament és bo i sense errors o errors que no són crítics.

El primer pas a seguir serà instal·lar la base de dades MongoDB en el servidor on volem que estigui localitzada. Un cop instal·lada la BBDD hem der poder-nos connectar des de Python a la BBDD. Per connectar-nos hem utilitzat una llibreria de Python anomenada Pymongo [8].

La llibreria Pymongo en permetrà connectar-nos a la BBDD de Mongo simplement introduint la direcció on està ubicada la base de dades (prèviament engegada i esperant peticions), també ens permetrà administrar la base de dades poden així crear col·leccions i introduir documents en aquestes col·leccions.

Després de tenir la base de dades creada i la llibreria Pymongo instal·lada, només ens farà falta emmagatzemar els *tweets*. Primer de tot hem de crear la col·lecció amb el nom del *hashtag* que volem, després agafarem els *tweets* en format JSON i guardarem les dades que ens interessen en un diccionari, per després guardar aquest diccionari com un document de la base de dades. Aquest procediment ho farem amb cada un dels *tweets* i així tindrem una col·lecció plena de *tweets* per poder-los analitzar en la propera fase.

Les dades que hem agafat de cada un dels *tweets* són: la id, el text del *tweet*, número de *retweets*, número de favoritos i idioma. Es podran afegir diferents tipus de dades si en el transcurs del projecte es necessiten.

5.2 Anàlisis de sentiments

Durant la fase de l'anàlisi de sentiments, hem realitzat dos dels objectius del projecte, un de molt important i un altre de secundari. El primer objectiu és analitzar el sentiment dels *tweets* per poder-los classificar segons la polaritat que tenen (positius, negatius o neutres). El segon objectiu es tracta d'analitzar l'impacte que te la ironia en la fase d'anàlisi i com es podria intentar evitar.

Per realitzar el primer objectiu, valorarem diferents mètodes d'anàlisi per després comparar els resultats entre ells i així poder valorar quin mètode és el més adient pel projecte.

Analitzarem tres mètodes diferents per a realitzar l'anàlisi de sentiments. A la hora d'escollir aquests tres mètodes, ens hem basat en el suport que tenen a internet, que siguin uns mètodes madurs, que es puguin utilitzar mitjançant Python i que no suposin un cost econòmic pel projecte.

5.2.1 NLTK API

NLTK és una plataforma molt utilitzada en l'entorn de l'anàlisi de sentiments [9]. Proporciona diferents funcionalitats per poder utilitzar més de 50 corpus i recursos lèxics juntament amb un conjunt de biblioteques de processament de text per classificar-lo, tokenitzar-lo, analitzar-lo,

mostrar el raonament semàntic. Podem dir que és la plataforma líder en el processament de llenguatge natural.

Per el nostre projecte, nomes utilitzarem una funcionalitat de la plataforma. Aquesta funcionalitat es diu VADER [10] i s'encarrega de calcular la intensitat i la polaritat d'un text donant com a resposta un valor entre -1 i 1. Els valors negatius ens indican que la polaritat del text es negativa i com més a prop del -1 estigui, més negatiu serà el sentiment del text. Passa el mateix amb la polaritat positiva, com més a prop estigui el valor de 1, més positiu serà el sentiment. En el cas de que la polaritat sigui neutral, el valor que té associat serà el 0.

En aquest cas no hem de crear cap classificador, ja que la pròpia eina en porta un de incorporat. El que fa aquesta eina per analitzar el text i arribar en un resultat, és analitzar las paraules que formen el text i les compara amb un diccionari per saber si les diferents paraules del text, estan en el diccionari. Aquests diccionari conté un conjunt de paraules juntament amb la seva polaritat i la seva intensitat.

La intensitat de cada text no només es calcula amb la intensitat de cada una de les paraules que estan en el text, sinó que te en conte altres aspectes també. La intensitat de cada paraula pot incrementar-se si esta escrita en majúscula, si la frase acaba amb un signe d'exclamació o amb ambdós casos al mateix moment.

Aquesta eina també és capaç de reconèixer emoticones i de analitzar frases que són positives i negatives a la vegada, donant com a resultat una mitja entre les dues polaritats, és a dir, si la part positiva de la frase és més intensa que la negativa, aleshores ens mostrarà com a resultat que és positiva però amb una intensitat més baixa, ja que es té en compte la part negativa del text. Com podem veure, aquesta plataforma és una gran candidata pel nostre projecte, ja que té tot el que necessitem per a classificar els *tweets* de diferents pel·lícules segons la seva polaritat.

5.2.2 Alchemy API

Alchemy API és una llibreria que forma part de la companyia IBM, que utilitza la tecnologia de processament de llenguatge natural i algoritmes d'aprenentatge automàtic per extreure metadades semàntics de contingut com ara informació sobre persones, fets, relacions, llengües, entre altres.

També ens proporciona una funcionalitat en la que ens pot dir la polaritat d'un text, però en aquest cas només ens donarà com a resultat la polaritat sense contar la intensitat, és a dir, ens dirà si es positiu, negatiu o neutre.

Aquesta llibreria és més senzilla que la eina mostrada anteriorment, però com que no té en compte la intensitat, podria ser que a la hora d'analitzar els *tweets* ens pugui donar un resultat més fiable o simplement que ho analitzi d'una forma diferent de la anterior.

5.2.3 TextBlob + NLTK: Creació d'un classificador

Per últim, per aprendre millor quin és el funcionament d'un classificador de text i per intentar d'adaptar-lo lo màxim al nostre projecte, hem creat un classificador utilitzant dues eines: TextBlob com a classificador i un corpus del NLTK per a entrenar aquest classificador.

TextBlob és una llibreria de python per a processar dades de text. Proporciona una API consistent per a processar llenguatge natural i ens dona funcionalitat varies com ara l'etiquetatge de text, anàlisis d'opinions, extracció de sintagmes nominals, entre altres coses.

En el nostre cas, hem utilitzat aquesta eina per a crear un classificador. Per poder entrenar aquest classificador, l'hem entrenat amb diferents dades que ens proporciona la plataforma NLTK. Per fer aquest entrenament hem agafat el corpus "Movie_Reviews" [9], que es tracta d'un conjunt d'opinions de persones sobre diferents pel·lícules i cada una de les opinions està marcada com a positiva o negativa.

El que fa el classificador a la hora de l'entrenament és agafar cada una de les paraules que conté una opinió, i marcar quina és la probabilitat que te una paraula d'estar en una opinió positiva o negativa, és a dir, si el classificador ha trobat 5 vegades una paraula i tots els cops ha sigut en una opinió positiva, les probabilitats de que una opinió amb aquella paraula sigui positiva són molt grans.

Després de fer l'entrenament del classificador, s'ha de fer el testeig per saber quin és el percentatge d'encert que té el classificador. Com que el corpus que hem utilitzat està compost per 1000 opinions positives i 1000 de negatives, hem utilitzat 800 de cada una per a fer l'entrenament i 200 de cada un per fer el testeig. La precisió que ens ha sortit a la hora de fer el test, ha sigut d'un 73,5%.

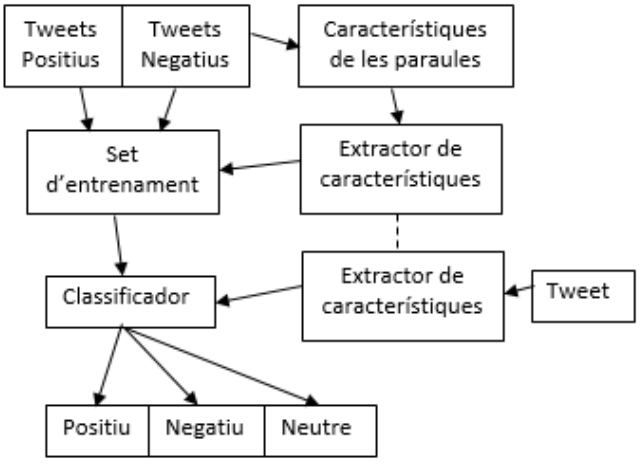

Fig.3: Resum del funcionament del classificador.

Un cop acabat el testeig, ja podem utilitzar-lo. Aquest classificador no és tant complet com ho són els classificadors de les dos eines anteriors, però com que l'entrenament que li hem fet ha sigut sobre l'opinió que ha tingut la gent sobre

una pel·lícula (el mateix tipus d'opinió que tenen els *tweets* del nostre projecte), pot tenir alguna possibilitat de fer millor l'anàlisi que les eines anteriors.

A la Figura 3 podem veure un resum del funcionament del classificador del que hem estat parlant anteriorment. Podem observar tots els elements dels que hem parlat i quin es el flux de dades que hi ha en el classificador.

5.2.4 Comparativa entre mètodes

Per saber quin és el mètode que utilitzarem en el projecte, farem una prova per veure quina eficàcia tenen a la hora de classificar cada un dels *tweets*. Per realitzar aquesta proba primer de tot agafarem *tweets* en un mateix idioma (agafarem l'anglès, ja que el nostre classificador esta només preparat per classificar text en anglès). Hem agafat 50 *tweets* en anglès que donen una opinió sobre la pel·lícula "El libro de la selva (2015)" i l'hem classificat manualment segons si era una opinió positiva, negativa o neutre.

Un cop classificats els *tweets*, hem posat en marxa cada una de les eines perquè ens classifiquin els *tweets* segons la seva polaritat i hem comparat els resultats de cada un dels mètodes d'anàlisi amb el que hem fet manualment. Així hem pogut obtenir el percentatge de resultats iguals que han obtingut cada un dels mètodes respecte la classificació manual.

Al veure els resultats de la proba que tenim a la Taula 3, podem descartar el classificador que hem creat (34%), ja que el percentatge de similituds amb la classificació manual és molt baixa. Si mirem els resultats que ha donat el classificador, hem vist que una gran part dels *tweets* els ha classificat com a neutres quan era bastant evident que eren o positius o negatius, això ens pot indicar que el problema que ha tingut el classificador és que amb l'entrenament que li hem fet, no l'hem enriquit de paraules com ens pensàvem en un principi i això ha comportat en que molts cops no ha identificat paraules que eren positives o negatives i les ha classificat com a neutres.

Els altres dos mètodes passen del 70% i el NLTK inclús passa del 75%. Podem dir que son bons resultats tenint en compte que analitzar la polaritat d'un text és molt difícil. Com que el mètode NLTK és més complert i ha tret més bon resultat que el Alchemy, aleshores agafarem aquesta opció perquè ens analitzi els *tweets* en el nostre projecte.

Prepararem el mètode NLTK perquè es connecti a la base de dades MongoDB on estan les diferents col·leccions amb els *tweets* a analitzar, analitzi cada un dels *tweets* i afegeixi dos camps per a cada *tweet*: un camp on es guardarà la polaritat del *tweet* i un altre on es guardarà la intensitat que te el *tweet*. Un cop analitzats tots els *tweets*, es guardaran de nou a la base de dades amb els dos camps nous, així tindrem totes les dades preparades per poder passar a la següent fase del projecte.

També guardem les característiques globals de l'anàlisi de la pel·lícula, és a dir, guardarem en una altre col·lecció el numero de *tweets* positius, neutrals i negatius, així com el total de *tweets* i el total de la suma de *tweets* positius i negatius (a la part de resultats s'explicarà el perquè).

TAULA 3
RESULTAT DE LA COMPARATIVA ENTRE MÈTODES

	NLTK	**Alchemy**	**TextBlob + NLTK**
Percentatge	76%	70%	34%

5.2.5 Ironia en l'anàlisi de sentiments

Per analitzar l'impacte de la ironia en l'anàlisi de sentiments, hem de saber què és la ironia i quines característiques te. La ironia és una figura literària el qual vol donar a entendre alguna cosa molt diferent o inclús contraria del que realment s'ha escrit.

Com hem vist abans, l'anàlisi de sentiments no te 100% de precisió a la hora de classificar textos segons la seva polaritat, hi ha diferents factors que fan que aquest percentatge baixi. Un d'ells és la ironia. L'impacte que te pot arribar a ser més important del que creiem, ja que la gent que sol utilitzar la xarxa social Twitter utilitzen molt bé la riquesa i complexitat del nostre idioma i del nostre llenguatge. Avui en dia s'utilitza molt la ironia en aquest tipus de xarxa, això implica que li haguem de prestar atenció per intentar disminuir el seu impacte per poder obtenir una informació més exacte a la hora d'analitzar els *tweet*s.

El problema que hi ha a la hora de analitzar un text irònic és que s'analitza el que es mostra i no te en compte altres tipus d'interpretació, això implica que ells textos irònics es puguin classificar amb una polaritat que no és la seva, i fa que els resultats no siguin tant precisos com haurien d ser.

No hi ha cap mètode que pugui analitzar i classificar textos irònics amb una precisió del 100%, però si que es pot detectar aquella ironia que no sigui molt subtil i que es pugui detectar amb algun patró, per exemple, quan un text diu alguna cosa molt positiva i acaba amb punts suspensius: "Una pel·lícula divertidíssima...".

També hem de pensar que inclús entre persones poden no estar d'acord en si una frase és o no és irònica. Si ni les persones coincidim al 100% en aquest tema, és molt difícil que una màquina pugui arribar a detectar els cassos que no siguin molt evidents, així que ara per ara, com a molt podrem minimitzar-la en els casos més clars.

5.3 Reporting

Un cop finalitzada la fase d'anàlisi, donarem lloc a la fase de *reporting* dels resultats per a realitzar els dos últims objectius del projecte, aquests són: Interfície senzilla on es mostraran els *tweets* i els resultats de la anàlisis de sentiments amb un *hashtag* en concret, i Incloure característiques dels usuaris a la hora de mostrar els resultats.

Per resoldre els dos objectius hem fet dos tipus de *report*, un on es mostraran els resultat de l'anàlisi de sentiments i un altre on es mostraran els *tweets* més representatius amb les característiques més rellevants dels seus usuaris.

També hem de comentar, que no s'ha creat cap interfície senzilla per mostrar els *report*s, sinó que aquests seran mostrats en PDF un cop creats. Tant per mostrar els resultats de l'anàlisi com per mostrar els *tweets* amb els seus usuaris més rellevants que ha tingut l'anàlisi.

Per dur a terme el *reporting* del resultat de l'anàlisi, hem utilitzat dues eines que s'utilitzen en l'àmbit del Business Intelligence: Kettle i Pentaho. La primera l'utilitzarem per manipular les dades i la segona per mostrar aquestes dades en forma de gràfiques que siguin fàcils d'entendre. Per la segona part, mostrarem els usuaris i *tweets* més rellevants utilitzant la llibreria ReportLab de python.

5.3.1 Kettle: Data Integration

La eina Kettle forma part de la companya Pentaho i ens proporciona la capacitat de extraure, transformar i carregar dades (ETL) utilitzant un nou enfocament basat en les metadades.

Abans de poder mostrar les dades en un *report*, primer les hem de manipular per que siguin compatibles per utilitzar amb Pentaho i perquè es mostri el qui volem i amb el format que volem [11].

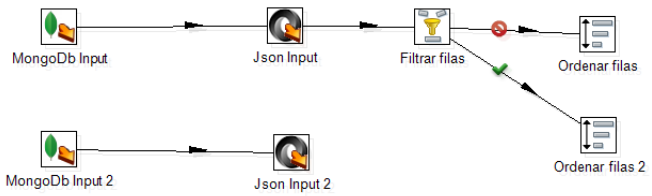

Fig.4: Flux de dades de la integració de dades.

Com podem veure a la Figura 4, en el primer pas de la primera transformació (MonoDB Input) carreguem les dades de MongoDB sobre la pel·lícula que haguem analitzat anteriorment. En el segon pas escollim quins són els camps que es podran mostrar en el *report* i en quin format han de estar per que sigui correcte.

En el tercer pas fem un filtratge mirant quina és la polaritat de cada *tweet*. Si els *tweets* son positius o negatius, aniran al pas "Ordenar Filas" i si són negatius aniran al pas "Ordenar Filas 2"(a l'apartat de resultats s'explicarà perquè es filtren els negatius). En l'últim pas simplement ordenem les dades segons la intensitat del *tweet*.

En la segona transformació, carreguem les dades de MongoDB amb les característiques principals de la pel·lícula a analitzar, i com hem fet amb la primera transformació, escollim quins són els camps que volem i amb quin format han d'estar.

Un cop creada i executada la ETL, ja tenim les dades tal i com les necessitem per poder crear els gràfics amb el resultat de l'anàlisi.

5.3.2 Pentaho: Report designer

Són un conjunt de eines de codi obert que permeten la creació d'informes relacionals i d'anàlisi. Per dur a terme aquest informe d'anàlisi, utilitzarem les dades resultants de la ETL anterior.

Per a mostrar el resultat de l'anàlisi, hem fet dos gràfics: un per mostrar la intensitat dels diferents *tweets* i un altre per mostrar la polaritat dels *tweets*. Així podrem saber quina es la quantitat de persones que els ha agradat o no la pel·lícula i quina ha sigut la intensitat d'aquest sentiment/opinió.

Pel crear el gràfic on es mostra la intensitat, em utilitzat les dades de l'anàlisi de cada un dels *tweets*. A l'eix Y tenim els diferent graus de intensitat mostrats des de el valor 100 al -100 i a l'eix X es mostra cada un dels *tweets*. Podem veure el resultat a la Figura 5.

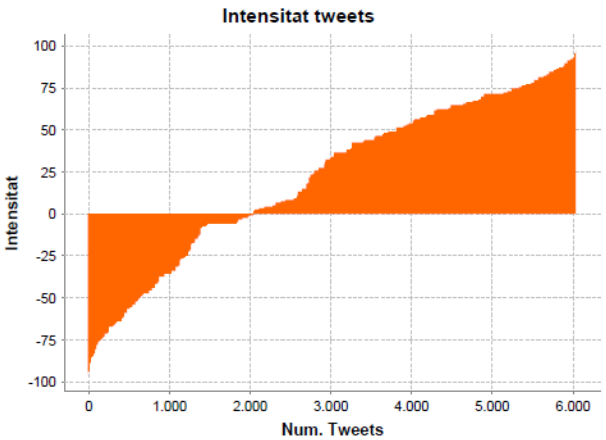

Fig.5: Gràfic de la intensitat dels *tweets*.

Per crear el gràfic on es mostra la polaritat, simplement hem agafat les dades globals de la pel·lícula (Transformació 2) i hem mostrat al eix Y la quantitat de *tweets* i en l'eix X una barra pels *tweets* positius i una altre pels negatius. Podem veure el resultat a la Figura 6.

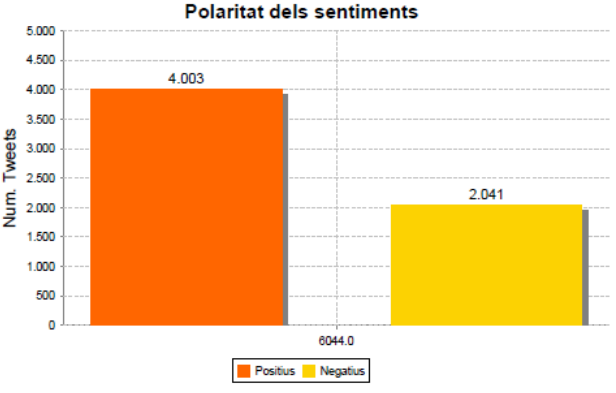

Fig.6: Gràfic de la polaritat dels *tweets*.

5.3.3 ReportLab

ReportLab és una llibreria que ens permet generar documents atractius i totalment a mida en PDF d'una forma ràpida i senzilla. Es poden fer documents totalment personalitzats i de gran qualitat [12].

Hem escollit aquesta llibreria per mostrar els *tweets* més rellevants perquè es poden crear documents i modificar-los d'una forma molt senzilla y ràpida i amb bona qualitat a la mateixa vegada. No hem utilitzat Pentaho, ja que ens ha resultat molt més difícil poder mostrar aquest tipus de dades i no tenien la qualitat esperada.

Per mostrar els *tweets* més importants, hem connectat amb la base de dades de la pel·lícula que volem fer el *report*, i hem agafat tots el *tweets* amb la seva polaritat i intensitat pertinent. Aleshores simplement hem creat el *report* amb la llibreria i ordenat els *tweets* segons els que volem mostrar (els més positius, els més negatius, els que tenen més *retweets*,...). Un cop ordenats, posem els que volem mostrar al *report*. A l'Apèndix 1, podem veure un exemple de *report* dels tweets més positius de la pel·lícula Civil War.

6 Resultats

Un cop finalitzat el desenvolupament del projecte, és el moment de analitzar els resultats que ens ha donat aquest per saber si realment el projecte fa el que ens indica l'objectiu principal. En aquest apartat ens fixarem amb el resultat de l'anàlisi utilitzant els *reports* que hem obtingut fent la ultima fase del projecte.

Per posar en marxa el projecte, hem recol·lectat 10.000 *tweets* sobre la pel·lícula "Capitán América Civil War" utilitzant el hashtag "#CivilWar". Hem filtrat aquests *tweets* per només obtenir-ne els que estiguin amb anglès. Un cop acabada aquesta fase, podem observar que tenim a la base de dades una col·lecció anomenada CivilWar on estan els 10.000 *tweets* que utilitzarem a les següents fases.

Un cop preparats tots els *tweets*, analitzem la seva polaritat i intensitat utilitzant el mètode NLTK que hem escollit a la fase d'anàlisi, i com a resultat ens dona una col·lecció anomenada CivilWarNLTK on els *tweets* tenien els camps intensitat i polaritat. Un cop finalitzat, podem analitzar el resultat que ens ha donat l'anàlisi de sentiments.

Primer de tot podem veure quina es la classificació que ha fet el mètode NLTK. La podem veure a la taula 4.

TAULA 4
RESULTAT DE LA CLASSIFICACIÓ DE *TWEETS*

	Positius	Negatius	Neutres	Total
Tweets	4.003	2.041	3.956	10.000

Podem veure que hi ha el doble de positius i neutres que de negatius, però sabent que és una pel·lícula que ha agradat molt al públic i que ha obtingut bones crítiques, podem dir que és un resultat coherent.

Com que el número de *tweets* neutres és molt elevat, hem mirat el contingut d'aquests *tweets* per saber millor quin tipus de *tweet* s'ha classificat com a neutre. Ens hem trobat que un percentatge molt gran dels *tweets* neutres era publicitat sobre la pel·lícula per anar a veure-la al cinema o simplement eren *tweets* de companyies on donaven informació de la pel·lícula, com ara els diners recol·lectats per aquesta.

Com que aquest tipus de *tweet* no ens dona cap tipus d'informació sobre l'opinió dels usuaris respecte la pel·lícula, aleshores hem decidit que els *tweet*s amb polaritat neutre realment no ens donaven informació útil, així que no els mostrarem en els *reports*. Només mostrarem els *tweets* positius i negatius que són aquells que ens donen informació respecte la opinió d'una persona amb una pel·lícula.

De la mateixa forma que hem analitzat el contingut dels *tweets* neutres, també ho hem fet amb els positius i amb els negatius. En els positius ens podem trobar una mica de tot: una gran part dels *tweets* són sobre gent que ha opinat positivament una pel·lícula, però també ens podem trobar que hi hagi algun *tweet* de publicitat que l'hagi classificat com a positiu, no ni han gaires, però ni han. També hem trobat algun cas en el que s'estava fent ús de la ironia i a la hora de classificar-ho no ho ha detectat.

Tot i així, el percentatge de encerts és bastant elevat (no s'han mirat els 4.000 *tweets* positius, sinó una petita part), un percentatge similar al que ens havia donat en la comparació en la fase d'anàlisis de sentiments.

En el cas dels *tweets* negatius, hem pogut veure que hi han menys errades, ja que no hem trobat gairebé cap *tweet* publicitari (pot ser degut en que en la publicitat sempre es parla bé del producte, en aquest cas la pel·lícula). De totes formes ens hem trobat algun cas en el que s'ha fet una mala classificació, per exemple, un dels *tweets* més negatius realment és una opinió positiva, però com que al principi utilitzava unes paraules mal sonants en majúscula (s'incrementa la intensitat), tot i que eren una expressió i no anaven referides a la pel·lícula, han fet que classifiques aquest *tweet* d'una forma errònia.

Per últim ens fixarem en la Fig.4 que ens mostra la intensitat dels *tweets*, és una gràfica interesant perquè ens mostra la intensitat del sentiment d'aquella persona cap a la pel·lícula. Si analitzem la gràfica, podem veure que la intensitat en els *tweets* positius són una mica més intensos que els negatius, això ens pot donar una informació bastant rellevant. Podria sorgir el cas de que tinguéssim aproximadament el mateix nombre de positius i negatius, però que en el cas del positius fossin molt intensos i en el dels negatius que ho siguin poc, això ens indicaria que a la gent que li ha agradat la pel·lícula, li ha agradat realment aquesta pel·lícula, en canvi a la gent que no li ha acabat de fer el pes, realment no esta fent una crítica molt dura de la pel·lícula. En conclusió, ens dona un altre tipus d'informació que pot servir a una empresa per estratègies futures.

Tot i algunes errades a la hora de classificar els *tweets*, el projecte es capaç d'analitzar *tweets* i mostrar els resultat d'una forma eficaç i fiable dintre dels valors de fiabilitat que hi ha en l'àmbit de l'anàlisi de sentiment. El projecte realitza el que es demanava en els objectius i es podria millorar en un futur, per poder arribar a un grau de fiabilitat més alta i per poder intentar personalitzar-lo més en l'àmbit cinematogràfic.

7 Conclusió

Un cop ja hem vist com s'ha realitzat el desenvolupament de les diferents fases del projecte presentarem una sèrie de conclusions que a simple vista semblen bastant clares.

Amb la realització de la fase de recol·lecció i emmagatzematge de *tweets* hem sigut capaços de poder recol·lectar i filtrar una gran quantitat de *tweets*, gràcies a l'ús d'una base de dades NoSQL i controlant les limitacions de *tweets* que Twitter ens deixa recollir en un espai de temps utilitzant la llibreria Tweepy. En aquesta fase hem pogut fer la preparació de les dades necessàries per poder fer posteriorment un anàlisi de polaritat, on és important utilitzar una gran quantitat de dades per poder obtenir uns resultats lo més fiables possibles.

A la segona fase hem comparat diferents mètodes per analitzar la polaritat dels *tweets* recol·lectat per així poder utilitzar el model que més ens convingui en el projecte, també hem creat un classificador per poder entendre d'una millor forma el funcionament real que te a la hora de classificar els *tweets*. Això ha comportat que puguem analitzar els resultats de l'anàlisi d'una forma més detallada, ja que sabem el perquè s'ha pogut classificar un *tweet* bé o malament.

Per últim hem sigut capaços de fer diferents *reports* de l'anàlisi utilitzant alguna de les eines que més s'utilitzen en el món del Business Intelligence, perquè al cap i a la fi aquests projecte forma part del Business Intelligence, ja que es tracta de l'anàlisi de dades per convertir-la en informació i així poder fer estratègies a curt i llarg termini segons la informació que haguem extret.

7.1 Possibles extensions

Poden haver-hi possibles extensions a les diferents fases del projecte. A la primera fase, una possible extensió seria la recol·lecció de dades diferents, és a dir, dades de diferents xarxes socials per així no centrar-nos nomes en Twitter.

En la segona fase, es poden introduir més extensions, com ara crear un classificador personalitzat i que sigui eficaç per l'àmbit cinematogràfic. Una altre extensió seria la categoritzar les dades segons el seu tipus, és a dir, categoritzarles segons siguin opinions, publicitat, SPAM o altres tipus. Així podríem agafar només el tipus de dades que necessitéssim per fer un anàlisi i tenir un resultat més precís.

Per últim, en la ultima fase, es podria fer *reportings* interactius amb l'ús de dashboards, per així poder mostrar els resultats d'un anàlisi d'una forma més amigable i de moltes formes diferents.

Agraïments

M'agradaria agrair a Jordi Casas Roma les iders i consells que m'ha donat durant el projecte. Totes les seves aportacions han fet possible el desenvolupament del projecte.

Bibliografía

[1] Brandwatch, Brandwatch Analytics, «https://www.brandwatch.com/es/brandwatch-analytics/,» [En línea]. [Último acceso: 22 2 2016].

[2] Tecnimedios, Funcionament Socialmention. «http://tecnimedios.com/blog/social_media/que-es-socialmention-y-como-funciona/,» [En línea]. [Último acceso: 22 2 2016].

[3] Microsoft Project, «https://www.microsoftstore.com/store/msusa/en_US/cat/Project/categoryID.69407700,» [En línea]. [Último acceso: 10 03 2016].

[4] Tweepy, Documentació Tweepy., «http://tweepy.readthedocs.org/en/v3.5.0/,» [En línea]. [Último acceso: 15 4 2016].

[5] Twitter, API REST., «https://dev.twitter.com/rest/public,» [En línea]. [Último acceso: 15 04 2016].

[6] GenbetDev, Tipus de Bases de dades NoSQL., «http://www.genbetadev.com/bases-de-datos/bases-de-datos-nosql-elige-la-opcion-que-mejor-se-adapte-a-tus-necesidades,» [En línea]. [Último acceso: 16 04 2016].

[7] MongoDB, Documentació MongoDB., «https://docs.mongodb.org/manual/?_ga=1.141323380.798105977.1459701046,» [En línea]. [Último acceso: 16 04 2016].

[8] MongoDB, Pymongo., «https://api.mongodb.org/python/current/,» [En línea]. [Último acceso: 16 04 2016].

[9] NLTK API, «http://www.nltk.org/install.html,» [En línea]. [Último acceso: 16 04 2016].

[10] C. &. G. E. (. Hutto, «VADER: A Parsimonious Rule-based Model for Sentiment Analysis of social media text,» de *Eighth International Conference on Weblogs and Social Media*, 2014.

[11] Pentaho, Creació d'un report., «http://wiki.pentaho.com/display/BAD/Create+a+Report+with+MongoDB,» [En línea]. [Último acceso: 13 05 2016].

[12] ReportLab, «https://www.reportlab.com/docs/reportlab-userguide.pdf,» [En línea]. [Último acceso: 18 05 2016].

APÈNDIX
A1. Exemple de report amb ReportLab.

TWEETS MÉS POSITIUS

Id tweet:	732983291230474245	Idioma:	en
Text:	@cannibyeolism ok but in #civilwar #spoilers when they're like 'wow we're so old like man we're old lol lol jokes' I CAN'T IT'S SO DATED		
Sentiment:	positiu	Retweets:	0
Polaritat:	0.97	Favorits:	1

Id tweet:	733105087799779328	Idioma:	en
Text:	#CivilWar was fantastic! They finally did my boy Black Panther justice!! He was amazing! Loved every second he was on (Spidey was cool too)		
Sentiment:	positiu	Retweets:	0
Polaritat:	0.96	Favorits:	0

Id tweet:	732954977446940672	Idioma:	en
Text:	Amazing, I love all super heroes. It's not just a war #MCU #TeamCap #TeamIronMan #CivilWar		
Sentiment:	positiu	Retweets:	0
Polaritat:	0.96	Favorits:	2

Id tweet:	732515781863481344	Idioma:	en
Text:	I could punch so many more holes in that movie but it's still an awesome film! Best Marvel film yet! #CivilWar		
Sentiment:	positiu	Retweets:	0
Polaritat:	0.96	Favorits:	0

Id tweet:	731861034126999553	Idioma:	en
Text:	I LOVE talking @Marvel with my brother. He's my hero, so talking heroes with him is basically just the best thing ever!! #CivilWar		
Sentiment:	positiu	Retweets:	0
Polaritat:	0.96	Favorits:	1

A2. Funció de recol·lecció de tweets i emmagatzematge amb el problema de limitació solucionat

```
def recollirTweets(hashtag):
    import tweepy, time, sys

    con = connectarBD()
    analsis = con.Tweets
    coleccio = analsis[hashtag]

    CONSUMER_KEY = '******************'
    CONSUMER_SECRET = '**********************'
    ACCESS_KEY = '************************'
    ACCESS_SECRET = '**********************'
    auth = tweepy.OAuthHandler(CONSUMER_KEY, CONSUMER_SECRET)
    auth.set_access_token(ACCESS_KEY, ACCESS_SECRET)
    api = tweepy.API(auth)

    maxId = -1;
    tweetCount = 0;
    while tweetCount < 10000:
        try:
            print("Buscant tweets... ")
            if maxId <= 0:
                tweetsNous = api.search(q=hashtag, count = 100)
            else:
                tweetsNous = api.search(q=hashtag, count = 100, max_id = str(maxId - 1))
            if not tweetsNous:
                print("No hi ha més tweets")
                break
            for tweet in tweetsNous:
                if analsis.coleccio.find_one({"text":tweet.text}) is None:
                    if tweet.lang == "en":
                        coleccio.insert({"id": tweet.id_str, "text": tweet.text, "idioma": tweet.lang,
                                         "retweets": tweet.retweet_count, "favs": tweet.favorite_count})
                        tweetCount += 1
                        print (tweetCount)
                        if tweetCount >= 10000:
                            break
            maxId = tweetsNous[-1].id
        except tweepy.TweepError as e:
            print("some error : " + str(e))
            time.sleep(60 * 15)
            continue
    desconectarBD(con)
```