TwitterBot per a la generació de notícies

Eduard García Gómez

Resum– Des de sempre, les persones han sentit la curiositat i necessitat de conèixer i saber el perquè de les coses del seu voltant. Per aquest motiu es van crear els medis d'informació, que fins ara es distribuïen físicament, ja sigui en forma de diaris, revistes, llibres i altres fonts. Actualment, però, la tecnologia avança ràpidament i aquests medis d'informació s'estan substituint per portals web, en els que es recull i es distribueix tota la informació de manera més ràpida. L'objectiu d'aquest projecte és crear un portal web que permeti la recol·lecció automàtica de les notícies a través de la xarxa social Twitter i, d'aquesta manera, evitar la necessitat de tenir un manteniment manual de les notícies com podria ser, per exemple, disposant de periodistes que cerquessin les notícies i uns redactors que les escrivissin. Els usuaris d'aquest portal podran accedir-hi des de qualsevol dispositiu amb connexió a internet i podran registrar-s'hi i interactuar-hi.

Paraules clau– Twitter, Notícies, classificador, robot, portal web, Internet, categories, Python, php, HTML, MySQL, Machine Learning

Abstract– Historically, people have felt the curiosity and need to understand and know why things happen around her. For this reason they created the information media, unitil now they were distributed phisically, either in the form of newspapers, magazines, books and other sources. Today, however, technology is advancing rapidly and these media vehicles are being replaced by web portals,these web portals collect and distributes all information more quickly. The aim of this project is to create a web portal that allows the automatic collection of news via the Twitter social network, and thus avoid the need for manual maintenance of the news as it could be for example, providing journalists who search for news and some writers that write it. Users of this site may be accessed from any device with an internet connection and be able to register and interact with the web.

Keywords– Twitter, News, classifier, bot, website, Internet, categories, Python, php, HTML, MySQL, Machine Learning

1 INTRODUCCIÓ

DEs de fa molt temps les persones sentim la necessitat d'informar-nos del que passa al nostre entorn i actualment, amb les noves tecnologies, aquesta necessitat es veu incrementada. Per tal que la informació que rebem sigui l'adequada ens cal informar-nos a través de fonts d'informació fiables i que s'actualitzin constantment.
Actualment, els medis físics de comunicació de notícies (diaris, revistes, etc.) ja no tenen tant mercat com tenien abans, ja que requereixen una quantitat de diners i de temps de fabricació major i més treballadors; les empreses que els fabriquen necessiten més materials i personal durant aquesta fabricació i els usuaris consumidors han d'anar a les botigues per tal adquirir-los.
Per aquest fet, els portals web de notícies han anat guanyant molt terreny i s'han convertit en una de les fonts de notícies principals per a la majoria de la població. Tot i necessitar un manteniment i actualització constants, tenen més accessibilitat i, a llarg termini, no són tant costosos per a l'empresa.
L'objectiu d'aquest article és proposar un sistema que ens permeti recol·lectar informació automàticament de les fonts de notícies principals, classificar aquesta informació segons la categoria a la que pertany i mostrar-la en forma de notícies a l'usuari en un portal web accessible des de qualsevol dispositiu amb connexió a internet.

• E-mail de contacte: egg1483@gmail.com
• Menció realitzada: Tecnologies de la Informació
• Treball tutoritzat per: Jordi Casas Roma
• Curs 2016/17

2 OBJECTIUS

L'objectiu del projecte és tenir un sistema independent que integri els mòduls necessaris per poder resoldre l'enunciat del projecte: desenvolupar un sistema que de forma automàtica reculli les principals notícies que apareixen a la xarxa social "Twitter", classifiqui aquestes notícies en categories genèriques i les mostri a través d'un portal Web. Aquest enunciat es pot dividir en quatre apartats principals:

- Desenvolupament d'un recol·lector que reculli els tweets del grup de Twitter que es selecciona, se n'extregui la informació en forma de tweets i l'emmagatzemi a la base de dades.
- Desenvolupament d'un classificador que s'encarregui de classificar els tweets abans de pujar-los a la base de dades.
- Desenvolupament d'una base de dades en la que es pugui emmgatzemar tota la informació necessària, tant dels tweets recol·lectats i la informació per classificarlos com dels usuaris i els grups on recol·lectarem els tweets.
- Disseny d'un portal web que permeti visualitzar les notícies recol·lectades en forma de tweets, permeti als usuaris registrar-se per poder visualitzar les categories i es pugui adaptar a tots els dispositius amb connexió a internet i mides de pantalla; per tant, que sigui responsive.

En finalitzar el projecte, el portal de notícies, el mètode de recol·lecció i el mètode de classificació han d'estar en funcionament.

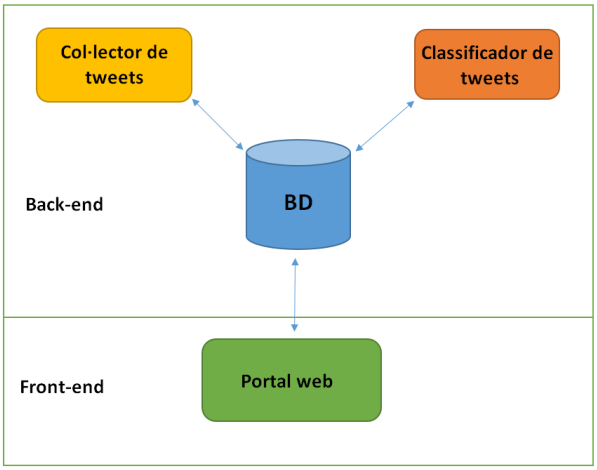

Fig. 1: Esquema dels mòduls

3 ESTAT DE L'ART

A dia d'avui a causa del gran desenvolupament de les tecnologies de la informació han aparegut moltes pàgines web de notícies. Algunes d'aquestes pàgines son dels mateixos diaris o revistes que han fet una versió digital per tal de poder arribar a més lectors.
Un dels grans al·licients i avantatges que tenen aquests portals de notícies web és que l'usuari, per tal de poder informar-se, no té la necessitat de desplaçar-se a un determinat lloc físic per poder adquirir i comprar el diari o la revista; solament requereix un dispositiu amb connexió a internet per accedir a la informació, suposant un gran estalvi de temps i diners per a aquest.
D'altra banda, els portals de notícies web tenen l'inconvenient de necessitar personal especialitzat en redactar i gestionar les notícies i cercar la informació constantment, fent que s'encareix-hi els costos de manteniment del portal. Per aquest fet, sorgeix la necessitat de disposar d'un sistema autònom que pugui recol·lectar les notícies per tal d'estalviar temps i diners i poder adaptar-se a la demanda dels usuaris.

4 PLATAFORMA DE DESENVOLUPAMENT

Fins al moment, les plataformes de desenvolupament que s'han utilitzat han estat:

- Visual Studio Code[7]: és un editor que facilita la programació en llenguatge web com HTML, PHP, Javascript i altres, ja que té plugins que ajuden a estructurar millor i programar més ràpidament, gràcies a poder treballar amb carpetes o a l'auto-completat de tags i funcions.
- MySQL workbench[8]: és una eina de MiSQL que facilita la gestió de la base de dades i permet monitoritzar-la fàcilment, incloent tot allò necessari per tenir el servidor de BBDD actiu.
- XAMPP[9]: és un paquet de programari lliure que inclou el servidor HTTP Apache, bases de dades MySQL i eines necessàries per poder executar PHP.

5 METODOLOGIA

La metodologia que s'ha emprat per dur a terme el projecta ha estat la metodologia SCRUM[6].

El mètode que es segueix és el següent: al final de cada setmana es miren els objectius marcats i es fa una avaluació dels que s'han complert i dels que no. Si hi ha algun objectiu que no s'ha completat, es miren les causes i s'introdueix en el pròxim sprint.

En l'assignació dels nous objectius per a la següent setmana, tenint sempre en compte la planificació general, s'assigna una prioritat a cada objectiu depenent de l'impacte en el projecte (Alt, Mitjà, Baix).

A més a més, la intenció és que es compleixin tots els objectius de l'sprint; per tant, es tenen en compte les altres tasques alienes al projecte que hi haurà durant la setmana (com ara exàmens, feines, etc) que podrien causar l'incompliment de diversos objectius.

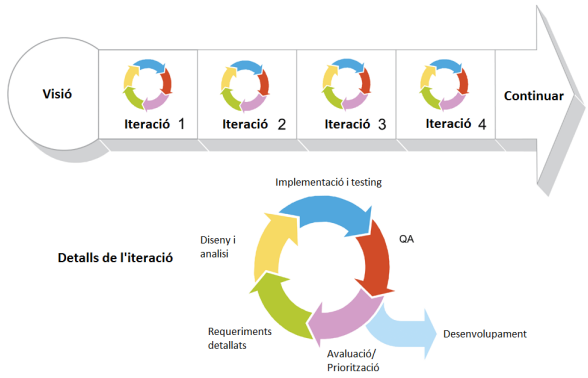

Fig. 2: Metodologia Scrum (Imatge adaptada de: scrumreferencecard.com/scrum-reference-card)

La planificació que s'ha seguit al llarg del projecte ha estat la seguent:

- **Fita 1 06/11/2016:** Tenir connexió amb Twitter i poder consultar tweets. Iniciar el desenvolupament de la web per poder mostrar els tweets. S'ha de tenir la base de dades feta amb els camps bàsics i enllaçada amb el mòduls corresponents.
- **Fita 2 18/12/2016:** Classificador funcionant correctament amb un rendiment acceptable. Pagines web responsive i amb els apartats de noticies. Base de dades refinada amb els camps necessaris. Sistema treballant correctament de forma conjunta.
- **Fita 3 22/01/2017:** Refinament de la classificació de noticies i investigació d'un mètode aplicant "machine learning" per classificar les noticies de forma més eficient. Millora de l'apartat d'administració i millora visual de la web.

Per al desenvolupament del projecte s'ha dividit el sistema en quatre mòduls principals: el recol·lector, el classificador, la base de dades i el portal web. Els mòduls de recol·lecció i classificació estaran governats des del portal web, on hi haurà un apartat d'administració que permetrà executar-los. Per tant, els mòduls no estaran tots junts sinó que es comunicaran a través de la base de dades, que és on hi haurà la informació.
Per a un bon desenvolupament del projecte es seguirà el paradigma Model Vista Controlador(MVC), un estil de codificació de qualitat, altament comentat i es dedicarà una part per poder fer testos dels diferents mòduls.

5.1 Base de dades

Finalment, la base de dades ha estat estructurada de la següent manera per poder proporcionar un bon protocol de comunicació de dades entre els mòduls del projecte.
L' estructura esta formada per 4 taules:

- Categories: conté el número identificador i el nom de cadascuna de les categories disponibles per a classificar els tweets.
- NoticeSources: és la taula on es guarden els canals de noticies i els hastags pels quals s'obtindran les notícies. Aquesta està formada per un identificador i el nom del canal o el hastag. Solament poden ser introduïts per els administradors del portal web, ja que son els que s'encarregaran de validar si és correcte o no abans.
- Tweets: conté tota la informació de cada tweets. Aquesta esta¡à formada per: L'identificador, que es un integer auto incremental que farà de clau primària de la taula, el l'objecte tweets sense modificar per si tinguéssim que tornar a buscar alguna informació o verificar-la de nou, el nom del usuari que l'ha publicat, la categoria que li assigna el classificador, la localització des d'on s'ha publicat en cas que estigui disponible la dada per poder fer futurs desenvolupaments amb geolocalització, l'idioma en que ha estat publicat per escollir els filtres de classificació adients i la data de creació. Totes aquestes dades venen directament de la petició REST feta des de l'API de Twitter[11] i no s'han de calcular.
- Usuaris: conté la informació dels usuaris del nostre portal web. D'aquests es guarda un identificador, un integer auto incremental que a la vegada fa de clau primària de la taula, el nom d'usuari, la contrasenya xifrada en MD5[12] per proporcionar una seguretat extra i evitar atacs de "man in the middle", el correu electrònic per poder enviar una clau de recuperació en cas de pèrdua, la data de creació del usuari i un camp booleà que ens indicarà si es administrador o no, el qual es fa servir en múltiples apartats de la web per proporcionar-hi accés només als usuaris administradors.

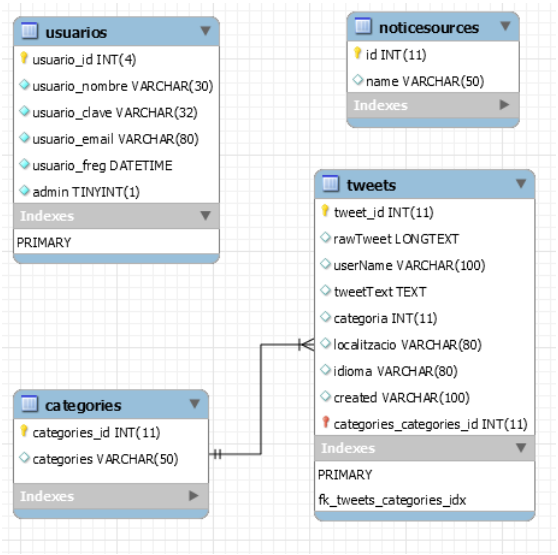

Fig. 3: Esquema taules BD

5.2 Recol·lector i classificador de Tweets

En primer lloc, el primer que s'ha de tenir en compte per començar és que a l'hora de construir el recol•lector s'ha de tenir present com s'obtindran els tweets. En aquest cas, Twitter ofereix dos mètodes per connectar-se a la seva xarxa i poder-los obtenir: API Rest i API Streaming. Les dues serien valides per fer aquest treball, però hi ha diferencies força grans entre elles.

- API REST[10]: és una API que s'utilitza per fer connexions volàtils amb el servidor; això significa que s'establirà la connexió, es realitzarà la petició, es rebran les dades i seguidament es tancarà la connexió i s'oblidarà. Això permet que hi hagi un baix consum de memòria i de tràfic de xarxa. Les dades son enviades en formar JSON, que es un format preestablert i que ocupa molt poc espai i permet tractar els missatges amb Javascript.

- STREAMING[13]: és una API que fa l'extracció dels tweets en temps real; això significa que està connectada contínuament amb Twitter a llarg termini, de manera que va obtenint actualitzacions constants dels canvis que hi ha (igual que un sistema de missatgeria instantània com pot ser el de Facebook). A nivell d'aplicació, per cada client connectat crea una connexió HTTP permanent amb Twitter; per tant, es pot deduir que tindrà un gran consum de memòria, CPU i xarxa a la part del servidor i, a més a més, que la velocitat de transferència estarà limitada al ample de banda màxim entre els dos punts de connexió, respecte l'API Rest.

Per al present projecte, degut als recursos dels que es disposa, la millor opció és utilitzar l'API Rest, ja que les consultes que es duran a terme es faran de forma única cada cert temps i s'emmagatzemarà la informació en el mateix moment en el que es rebi.
Per accedir a l'API Rest s'utilitzarà una llibreria existent anomenada Tweepy. Aquesta és una llibreria de fàcil ús i que permet fer la connexió amb Twitter de manera senzilla a l'hora de recol·lectar els tweets. A més a més, fa que s'estalviï temps de desenvolupament i que aquest es pugui dedicar a altres funcionalitats del projecte.

Per realitzar el mòdul del recol·lector de tweets s'ha creat diverses classes i funcions les quals de moment a falta d'una reestructuració son: classes Model i Controller i els metodes getTweets, uploadToDB i init tper poder executar el recol·lector des de la web al cridar al script de python.

5.2.1 Model

A la classe Model hi tenim els mètodes per la connexió i obtenció dels tweets. Per dur-ho a terme es requereix de l'autenticació OAuth[14].
L'autenticació OAuth consisteix en autenticar-se de forma segura utilitzant 4 parametres: consumer key, consumer secret, access token, acces token secret. Un cop tenim aquest 4 parametres els podem utilitzar per obtenir els objectes Auth i Api que, són els que ens permetran accedir a les funcions per obtenir els tweets.
Per poder fer la connexió i l'autenticació fa falta tenir un usuari creat a Twitter per poder vincular l'aplicació amb l'usuari i obtenir les claus. Seguidament, s'explicarà com es realitza la connexió: el primer pas és demanar a Twitter les claus com a administradors d'una aplicació prèviament registrada mitjançant la plataforma per a desenvolupadors. Un cop es tenen les claus, l'aplicació ja tindrà els mateixos permisos d'escriptura i lectura que se li hagin assignat a l'aplicació registrada a Twitter.

- Consumer key: és l'identificador del client, associada a l'aplicació de Twitter.

- Consumer secret: és la contrasenya del client que ens permet autenticar-nos davant del servidor de Twitter.

- Access token: és l'identificador que permet autenticar els privilegis que tindrà el client una vegada hagi passat el primer pas d'identificació.

- Acces token secret: és enviat amb l'acces token a mode de contrasenya per evitar mals usos en cas de conèixer només aquest últim.

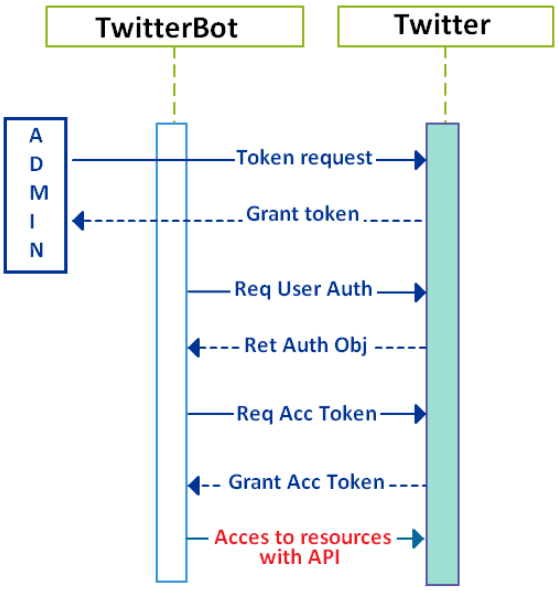

Fig. 4: Diagrama de seqüencia Oauth

Una vegada s'ha establert la connexió i s'ha autenticat, ja es pot procedir a obtenir els tweets amb el mètode:

- getTweets (arg): obté un objecte amb 10 tweets, per cadascun d'ells n'obté la informació i els classifica cridant al mètode Classificar() i finalment els puja a la base de dades utilitzant el mètode uploadToDB(). L'argument que se li passa a la funció és el nom del canal de tweets o un hastag i segons sigui un o l'altre fa servir una funció diferent de l'api de Twitter per obtenir els tweets.

5.2.2 Controller

La classe controller és la classe principal del recol·lector en la que hi ha el mètode init, que ens permet executar tot l'script des de la web i poder obtenir els tweets.

- init(): s'inicialitza tots els atributs del model i classificador amb les instàncies de les classes Model i Controller, es carreguen les llistes amb les paraules per poder classificar i es crida al mètode getTweets() per tal d'obtenir els tweets.

- uploadToDB(rawtweet, userName, TweetText, categoria, localització, idioma, created): és el mètode encarregat de carregar els tweets a la base de dades. Les dades que se li passa a la funció que s'obtenen del mètode

getTweets son: el tweets sense modificar, el nom d'usuari que l'ha publicat, el text del tweets, la categoria que es obtinguda del mètode classificador(), la localització, l'idioma en que s'ha escrit el tweets i la data de creació.

```
INSERT INTO
tweets(tweet_id,rawTweet,userName,
TweetText,categoria,localitzacio,
idioma, created)
VALUES
(%s,%s,%s,%s,%s,%s,%s,%s)
```

Codi. 1: Carregar informació tweet

- classificador(text): retorna una categoria segons les paraules que té el camp de text que se li passa. Per obtenir la categoria, es passa el text a minúscules i es separen les paraules. Una vegada es té la llista de paraules de text de twitter, es fa la comparació amb cada una d'aquestes per categoria, i cada vegada que es troba una concidència, s'augmenta el comptador en un. Finalment, es mira quin és el comptador més gran; en el cas que hi hagi una coincidència amb un o més que tinguin el mateix número, s'assigna a una categoria neutra.

```
Entretenimiento = ['Anime','Cine',
'Television','Pelicula','Musica',
'Videojuego','Deporte', ...]
```

Codi. 2: Tags classificador

5.3 Portal web

El portal web ha estat desenvolupat amb l'ajuda d'un framework d'estils anomenat Bootstrap, el qual permet fer que la web sigui responsive de forma nativa i que es pugui ajustar a diferents tipus de pantalles (per exemple, les dels dispositius mòbils). Per altra banda s'han utilitzat els llenguatges HTML, PHP i Javascript.

Aquest portal consta de diferents apartats:

- Home: mostra els últims tweets recol·lectats sense filtrar per categoria. Si s'accedeix amb un usuari es pot veure la classificació de les notícies per categories. En el cas que la persona que l'utilitza no recordi la contrasenya, la pot recuperar a l'apartat de recuperació de contrasenya; per altra banda, si una persona no està registrada es pot crear un compte. En el moment en el que es crea un nou compte, es demana el nom d'usuari, el correu electrònic i una contrasenya, que serà guardada de manera segura a la base de dades. Al registrar-se a la base de dades, l'usuari rep un identificador únic autoincremental, que serà el que el diferenciarà de la resta d'usuaris juntament amb el nom que també es únic. Una vegada s'ha accedit, es poden veure els tweets segons la categoria i accedir al perfil d'usuari. En el cas que la persona que hi accedeix sigui un administrador, permet accedir al panell de l'administrador.

- Panell d'administrador: permet recol·lectar tweets escollint el canal o el hastag que es utilitzar per recol·lectar. A més a més, permet afegir canals de notícies o hastags per cercar a la base de dades. Solament es pot accedir a aquesta pàgina si s'ha iniciat sessió amb un usuari administrador (en el cas de no ser-ho, l'accés és denegat). D'aquesta manera s'evita que es puguin dur a terme atacs en el cas de conèixer l'adreça de la pàgina d'administració.

- Perfil d'usuari: mostra la informació del usuari (nom, correu, data de registre) i permet modificar la contrasenya per una de nova en cas que sigui necessari.

5.3.1 Bootstrap

Bootstrap és un framework de codi obert, que permet fer el disseny de portals web més ràpidament i visualment més atractius una vegada s'ha après a utilitzar-lo. En el projecte ha estat una eina essencial per tal de poder desenvolupar la web i que fós responsive, ja que les webs creades amb boostrap ja ho són per defecte. D'altra banda, el seu aprenentatge és complex, ja que la forma de fer els elements clàssics d'HTML va canviant, i passa a ser un disseny basat en "contenidors" i classe a l'hora d'estructurar els portals web.

5.3.2 Estructura de fitxers

Per tal de tenir ben estructurat i diferenciada cada part de la web s'han separat els fitxers del servidor en diferents carpetes per tal de facilitar futurs desenvolupaments o, simplement, la modificació d'algun apartat. A l'arrel s'hi troben les pàgines principals de la web (home, administració, perfil d'usuari) i llavors hi ha cinc carpetes més (detallades a continuació):

- Css: s'hi troben els fitxers d'estils del framework Bootstrap i el fitxer de la web.

- Js: s'hi troben els fitxers principals Javascript de Bootstrap.

- Fonts: s'hi troben les fonts i gràfics que apareixen a la web. La majoria són de bootstrap, cosa que permet que siguin escalables a qualsevol mida de pantalla, i el resultat es veu més professional.

- sistemaUsuarios: s'hi troben tots els fitxers php que s'utilitzen per gestionar les operacions que fan els usuaris. Entre els fitxers, hi ha els que gestionen l'accés a la base de dades, el canvi de contrasenya, la comprovació de l'usuari al iniciar sessió, el logout, la recuperació de contrasenya i el registre. Aquests fitxers no són pàgines en si, sinó components d'aquestes i, per incloure'ls, s'ha de fer fent una recarrega d'una secció de la web amb AJAX[15] o fent la següent crida:

```
<?php
    include 'SistemaUsuarios/File.php';
?>
```

Codi. 3: Inclusió PhP

- TweetBotGN: és la carpeta principal on hi ha els arxius de Python que controlen la connexió amb Twitter i fan la recol·lecció i classificació dels tweets.

5.3.3 Seguretat web

Com a mesures de seguretat s'han implementat diverses solucions que ajuden a evitar atacs de "man in de middle", accés a pàgines restringides i per millorar l'autenticació de l'usuari.
Protecció de pàgines d'administració: una de les primeres coses que es prova quan s'intenta accedir al control d'una web de forma fraudulenta, és esbrinar els fitxers que porten a les pàgines d'administració de la web per accedir-hi a través de l'adreça web. Per tant, el que s'ha fet a estat restringir l'accés a aquestes pàgines utilitzant una part de codi que comprova que l'usuari que intenta accedir tingui permisos d'usuari administrador a la base de dades. El codi que ho implementa és el següent:

```
if(!sessioUsuari){
  #mostrar error d'usuari no loguejat
}else{
  if(valor admin == True){
    #mostrar pagina admin
  }else{
    #mostrar error d'usuari no admin
  }
}
```

Codi. 4: Acces pagina administració

Protecció de la contrasenya: un altre problema és a l'hora de crear la contrasenya de l'usuari i enviar-la a través d'internet, tant per identificar-se com per registrar-se, ja que si aquesta contrasenya no està xifrada viatjarà en clar i podria ser interceptada per alguna altra persona, és a dir, si la contrasenya és interceptada en clar podrà ser utilitzada de forma fraudulenta. A més a més, és possible que l'usuari que s'hagi registrat a la web utilitzi aquesta contrasenya en altres webs o per altres tràmits i gestions; això significa que es posa en risc la integritat d'altres comptes que pugui disposar l'usuari.
D'altra banda, si per algun motiu una persona pogués accedir a la base de dades i obtenir les contrasenyes sense xifrar, els responsables seriem nosaltres, i això podria portar problemes greus a nivell legal.
Per aquest fet, s'han xifrat les contrasenyes en MD5 quan s'envien a través de la xarxa i s'emmagatzemen una vegada a la base de dades; d'aquesta manera, es poden evitar atacs "man in de middle".

6 MACHINE LEARNING

Per tal de millorar la classificació dels tweets s'ha investigat un mètode de classificació que utilitza la intel·ligència artificial per anar aprenent i millorar la classificació al llarg del temps. Per dur a terme aquest apartat s'han utilitzat unes llibreries de Phyton ja existents anomenades "TextBlob"[16] i "NLTK"[17], que proveeixen les eines necessàries per poder tractar el Processament del Llenguatge Natural (NLP), ajudant-nos amb l'extracció de les paraules importants i l'anàlisi d'aquestes.

El primer pas per començar a construir el classificador és escollir l'algoritme adequat. En aquest projecte s'ha escollit el "Naive Bayes", ja és un algoritme que s'adapta força a les necessitats.

El següent pas és estrenar el classificador. Per fer-ho és necessari tractar les dades incloses, ja que les paraules que no aportin informació per classificar (adjectius, connectors o signes de puntuació) no es volen incloure. L'entrenament del classificador es pot fer des de diferents orígens, un arxiu "JSON", CVS o directament des d'un "Array" amb les frases positives i negatives. En aquest projecte s'ha realitzat des d'un fitxer "JSON" ja que permet actualitzar-lo amb les noves frases classificades de forma més rapida i es més fàcil de supervisar, degut a que es pot obrir de manera manual per veure'l.

```
[
 { "text": "celta asalta bernabeu
  madrid pierde segundo partido
  consecutivo", "label": "pos" },
 { "text": "trump: gusta tuitear
  pero unica forma tengo expresarme",
  "label": "neg" }
]
```

Codi. 5: Fitxer JSON

Per recol·lectar el text i que estigui adaptat al format correcte s'ha creat un recol·lector com el de tweets, on solament s'emmagatzema el text dels tweets i es tracta per adaptar-lo com s'ha dit anteriorment, sense puntuació i sense les paraules que no aporten informació rellevant.
Els passos principals per tractar el text són els següents:

- Css: s'hi troben els fitxers d'estils del framework Bootstrap i el fitxer de la web.
- Passar el tot text a minúscules.
- Treure els signes de puntuació.
- Crear un array amb les paraules del text.
- Eliminar les paraules irrellevants.
- Passar l'array a text novament.
- Incloure el text al fitxer JSON, amb la polaritat adequada segons la categoria que estiguem tractant.

Un altre pas que s'hauria de realitzar però no s'ha pogut dur a terme és passar cada paraula a la seva paraula d'origen. D'aquesta manera s'aconseguiria un rendiment molt més elevat ja que les paraules amb el mateix origen serien considerades les mateixes (rapidesa, rapidíssim, ràpidament $\longrightarrow$ ràpid)

Un cop s'ha aconseguit l'arxiu amb el format adequat ja es pot procedir a entrenar el classificador fent la crida següent:

```
with open('deportes2.json', 'r',
        encoding="utf-8") as fp:
  cl = NaiveBayesClassifier(fp,
```

```
        format="json")

cl = NaiveBayesClassifier(train)
cl.classify(text)
```

Codi. 6: Crides entrenar i classificar

Un cop entrenat el classificador, el següent pas es comprovar l'eficiència amb dades de test, que s'estructuren de la mateixa manera que les d'entrenament.

[(frase, polaritat), (frase, polaritat)]

En aquest projecte amb l'entrenament efectuat els resultats han estat d'un setanta per cent d'encert, fallant en dos casos per fals positiu. L'entrenament que es va fer per al classificador, va constar de 50 frases d'esports i 25 frases de temes que no eren d'esports totes elles extretes de Twitter i tractades com s'ha explicat als apartats anteriors.

Com es pot veure a la figura 5 també ens mostra les "característiques" més importants, aquestes son les paraules que tenen més pes a l'hora de decidir si una frase pertany o no al grup indicat.

En aquesta prova es les 3 paraules que tenien més pes eren:

- Partido : si no contenia la paraula "partido" tenia un pes d'1.4 de no ser positiva.
- Para: si no contenia la paraula "para" tenia un pes d'1.3 de ser positiva.
- Momento: si no contenia la paraula "momento " tenia un pes d'1.3 de ser positiva.

A l'apèndix s'adjunta una captura amb el codi de l'apartat de test i les frases classificades.

```
C:\Users\egarc\Desktop\TFG\textblob>python naive.py
neg
neg
pos
neg
neg
pos
pos
neg
pos
pos
Most Informative Features
       contains(partido) = False              neg : pos    =      1.4 : 1.0
          contains(para) = False              pos : neg    =      1.3 : 1.0
       contains(momento) = False              pos : neg    =      1.3 : 1.0
          contains(parte) = False              pos : neg    =      1.3 : 1.0
contains(deportescuatro) = False              neg : pos    =      1.2 : 1.0
          contains(barça) = False              neg : pos    =      1.2 : 1.0
      contains(bernabéu) = False              neg : pos    =      1.2 : 1.0
        contains(sergio) = False              neg : pos    =      1.2 : 1.0
       contains(chelsea) = False              pos : neg    =      1.1 : 1.0
          contains(gran) = False              pos : neg    =      1.1 : 1.0
None
0.7
```

Fig. 5: Resultats test machine learning

Finalment, s'ha de passar a l'entorn real de Twitter. Per poder classificar les frases s'ha de seguir el mateix procediment de tractament que s'ha comentat anteriorment ja que sinó al afegir-les al fitxer JSON incorporarien paraules o signes de puntuació que afectarien negativament a la precisió del classificador.

Després de fer proves en l'entorn real s'ha vist que els resultats obtinguts no són tant bons com els de test, ja que en molts tweets no hi ha suficient informació rellevant com per classificar-ho correctament. S'ha observat un altre problema i és que si la frase a analitzar no conté cap coincidència amb les paraules de les frases positives o negatives, aquesta es dóna com a positiva quan en realitat és possible que no ho sigui. Una manera per solucionar aquest problema seria tenir una mostra d'entrenament suficientment gran com per tenir exemples de frases de molts tipus, d'aquesta manera no es trobaria que no hi ha cap coincidència.

Així doncs, podem concloure que substituir l'actual classificador per aquesta implementació en l'estat actual no seria una millora, ja que no en faria una classificació millora i causaria molts falsos positius degut a que la mostra d'entrenament no és suficientment gran.

7 RESULTATS

Els resultats obtinguts finals son un sistema que funciona de forma quasi autònoma en el que tot es gestiona des de una interfície web que ens permet complir els objectius principals del treball: recol·lectar, classificar i mostrar tweets mitjançant una interfície web.

A continuació es mostren les diferents vistes de l'aplicació:

En la següent imatge podem apreciar com la web s'adapta a qualsevol mida de pantalla i a qualsevol resolució, permetent així una bona visualització en qualsevol dispositiu.

Fig. 6: Vista web responsive

En aquesta tenim la vista general del portal, on es pot veure les noticies i les categories, apart d'accedir als diferents apartats de la web.

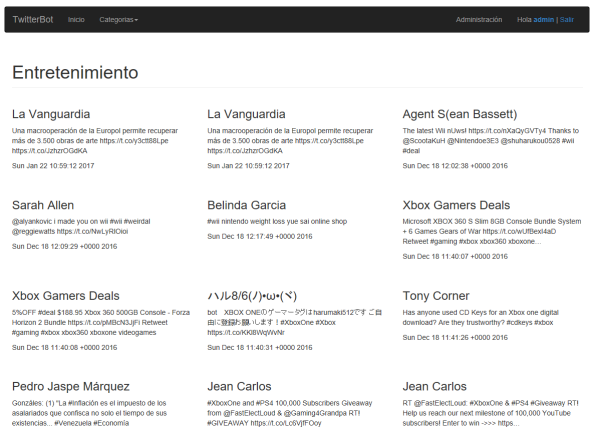

Fig. 7: Vista web

En el panell d'administració es pot veure els apartats per carregar notícies d'un canal, el d'introduir al sistema nous canals i el gestor d'administradors, que ens permet modificar els usuaris que tenen permisos per administrar.

Fig. 8: Vista web administració

En la següent imatge tenim el desplegable per fer accedir amb l'usuari, recuperar la contrasenya, o registrar-te al portal.

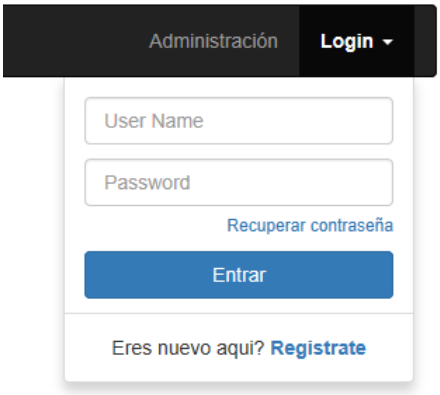

Fig. 9: Vista web Login

A l'última imatge tenim el formulari de registre on es mostra la informació que a d'introduir l'usuari per poder-se registrar.

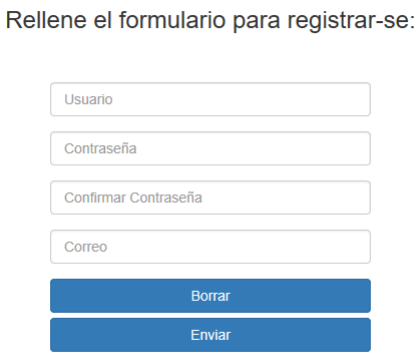

Fig. 10: Vista web registre

8 Conclusions Prèvies

Una vegada s'ha finalitzat el projecte s'ha pogut veure que s'han anat assolint els objectius que s'havien establert en un inici. Tot i haver aparegut algun imprevist durant les entregues inicials s'ha pogut recuperar correctament la planificació i complir-la amb els objectius, ja que l'ús del framework Bootstrap va facilitar la feina del desenvolupament de la web.

Pel que fa a la metodologia emprada hi ha algunes mancances ja que no hi ha un històric de les tasques realitzades i de les tasques pendents, però al tractar-se d'un projecte en el que solament hi havia un desenvolupador, i aquest era petit, això no ha suposat un inconvenient.

Els requisits funcionals proposats han estat completats amb un rendiment acceptable. La recol·lecció de tweets s'ha completat al cent per cent, tenint en compte que es poden recol·lectar tweets de diferents fonts; la classificació de tweets es trobaria al vuitanta per cent aproximadament, ja que la que s'ha aplicat finalment es podria millorar amb un mètode d'intel·ligència artificial com el que es va investigar, però la manca de temps no ha permés refinar-ho suficientment com per aplicar-lo en un entorn real; la web es trobaria al setanta per cent tenint en compte que és totalment funcional, però es podrien millorar aspectes visuals i d'administració.

En el seguent apartat s'exposaran les línies futures de desenvolupament que ajudarien a completar el projecte i que no han estat possibles de dur a terme ja que el temps de desenvolupament proposat no ho permetia. Aquestes millores suposarien un acabat de la web més professional i amb una qualitat superior al projecte, a part que ajudarien a millorar alguns aspectes que no s'han pogut tractar.

9 Línies futures de desenvolupament

Tenint en compte les característiques del present projecte, en un futur es podrien ampliar o millorar. Seguidament es presenten una sèrie de propostes de línies futures de desenvolupament que es podrien aplicar per aconseguir un portal web de millor qualitat:

Millora de l'aspecte visual. Tot i que la interfície web és correcta, es podria aplicar-hi un estil més personalitzat i proper al de Twitter. Per dur-ho a terme, s'afegirien elements decoratius, com ara icones, imatges de fons i plantilles, per a cada noticia; d'aquesta manera les notícies no tindrien una presentació plana com la que tenen actualment.

Millorar l'algoritme de classificació. L'algoritme actual de classificació proporciona uns resultats acceptables, però aquest no és el més idoni ja que s'han d'anar actualitzant manualment les paraules incloses a les llistes. Per tal de millorar l'algoritme, es podria aconseguir que anés millorant els paràmetres de classificació automàticament (autoaprenent) per tal de millorar els resultats de la classificació i que aquests siguin més precisos. Un apartat del present projecte s'ha dedicat a investigar un mètode de "machine learning" que seria útil, però per poder-ne treure el màxim rendiment s'hauria de dur a terme un altre projecte, ja que és molt complex.

Multi idioma. La llengua utilitzada a la web és la llengua espanyola, ja que és una llengua que té un nombre superior de parlants que la llengua catalana i, d'aquesta manera, la web pot arribar a un major nombre d'usuaris. Així doncs, una millora que es podria aplicar seria poder adaptar la web a diferents llengües (per exemple, el català, l'anglès, el francès, etc.). Per fer-ho, l'usuari triaria l'idioma en el que vol veure el portal i, gràcies al camp d'idioma dels tweets, es podria saber en quin idioma estaria redactat i poder-lo traduir al idioma seleccionat, o simplement només mostrar-los; d'aquesta manera s'aconseguiria que el portal s'internacionalitzés.

Afegir un fòrum per obtenir feedback dels usuaris del portal. Quan s'obtingués feedback dels usuaris del portal, aquests proporcionarien informació que els desenvolupadors de la web no veuen per poder millorar la web. A més a més, també podrien informar d'errors i es podria fer un sistema de privilegis per als usuaris, per tal que es sentissin més integrats i motivar-los a participar amb la web. Entre els privilegis hi hauria la possibilitat de poder administrar algun apartat de la web, com ara l'administració de les notícies, veure si estan classificades correctament, que no n'hi hagués d'irrellevants o de contingut no apropiat, etc.

10 Agraïments

Agrair al tutor Jordi Casas-Roma per guiar-me durant la realització del projecte, així com les propostes de millora i l'atenció donada. També agrair a la meva parella per tota l'ajuda moral i entendre que no es pot fer tot quan hi ha feina a fer.

Referències

[1] Twitter.com [Lloc web]. [Consulta: 25 de setembre 2016] Disponible: https://twitter.com

[2] W3Schools[Lloc web]. [Consulta: 25 de setembre 2016] Disponible: http://www.w3schools.com/

[3] Bootstrap[Lloc web]. [Consulta: 25 de setembre 2016] Disponible: http://getbootstrap.com/

[4] Scikit-learn[Lloc web]. 2013. Scikit-learn. [Consulta: 25 de setembre 2016] Disponible: http://scikit-learn.org/stable/

[5] Mongo DB Manual [Lloc web]. 2016. MongoDB. [Consulta: 25 de setembre 2016] Disponible: https://docs.mongodb.com/manual/

[6] SCRUM[Lloc web]. 2016. Metodologia Scrum. [Consulta: 4 de Novembre 2016] Disponible: https://es.wikipedia.org/wiki/Scrum

[7] Visual Studio Code[Lloc web]. 2016. Microsoft. [Consulta: 4 de Novembre 2016] Disponible: https://code.visualstudio.com/

[8] MySQL Workbench[Lloc web]. 2016. Oracle. [Consulta: 4 de Novembre 2016] Disponible: http://www.mysql.com/products/workbench/

[9] XAMPP[Lloc web]. 2016. Apache Friends. [Consulta: 10 de Desembre 2016] Disponible: https://www.apachefriends.org/es/index.html

[10] REST[Lloc web]. 2016. Twitter. [Consulta: 20 de Novembre 2016] Disponible: https://dev.twitter.com/rest/public

[11] API Twitter[Lloc web]. 2016. Twitter. [Consulta: 20 de Novembre 2016] Disponible: http://www.mysql.com/products/workbench/

[12] MD5 php. 2016. PHP. [Consulta: 25 de Novembre 2016] Disponible: http://php.net/manual/es/function.md5.php

[13] STREAMING[Lloc web]. 2016. Twitter. [Consulta: 25 de Novembre 2016] Disponible: https://dev.twitter.com/streaming/overview

[14] Oauth[Lloc web]. 2016. Twitter. [Consulta: 25 de Novembre 2016] Disponible: https://dev.twitter.com/oauth

[15] Ajax Workbench[Lloc web]. 2016. JQuery. [Consulta: 25 de Novembre 2016] Disponible: http://api.jquery.com/jquery.ajax/

[16] TextBlob[Lloc web]. 2016. TextBlob. [Consulta: 28 de Desembre 2016] Disponible: https://textblob.readthedocs.io/en/dev/

[17] Natural Language Toolkit[Lloc web]. 2016. NLTK. [Consulta: 28 de Desembre 2016] Disponible: http://www.nltk.org/

APÈNDIX

Codi test Machine Learning

```
from textblob.classifiers import NaiveBayesClassifier
train = [
    ('celta asalta bernabéu madrid pierde segundo partido consecutivo', 'pos'),
    ('real sociedad horario partido copa ante barça', 'pos'),
    ('sergio ramos sevillista morancos repasan actualidad sevilla', 'pos'),
    ('murray gana pero también lesiona tobillo muguruza pasa ronda open australia', 'pos'),
    ('mourinho como loco jugador liga', 'pos'),
    ('luis enrique rueda prensa pierdas últimas declaraciones deportescuatro', 'pos'),
    ('sergio araujo entrenamiento palmas', 'pos'),
    ('barça despide joey dorsey criticar servicios médicos club', 'pos'),
    ('renovación messi deportescuatro entrevista', 'pos'),
    ('partido bernabéu', 'pos'),
    ('ideas para reforma laboral parte sanciones extinciones indemnizadas incumplimientos graves ', 'neg'),
    ('suecia supermercados reemplazan etiquetas marcas láser tecnología', 'neg'),
    ('votada 24h obama conmuta gran parte pena prisión chelsea manning liberada mayo', 'neg'),
    ('historia beluga aviones carga utiliza airbus para', 'neg'),
    ('precio electricidad tiene techo supera miles euros espera', 'neg'),
    ('cazador furtivo muere pisoteado elefantes otro queda herido gravedad', 'neg'),
    ('webcam graba momento exacto erupción volcán méxico', 'neg'),
    ('nueve relojes momento sistema navegación galileo fallado', 'neg'),
    ('verde mega planta solar elche campaña', 'neg')
]

with open('deportes2.json', 'r', encoding="utf-8") as fp:
    cl = NaiveBayesClassifier(fp, format="json")

cl = NaiveBayesClassifier(train)

test = [
    ('mala praxis momento exacto episodio hospital', 'neg'),
    ('precio electricidad nuves', 'neg'),
    ('comportamiento barça nefasto ante madrid', 'pos'),
    ('twitter deja miles dolares momento inflación', 'neg'),
    ('historia vaca muere extrañas circumstancias', 'neg'),
    ('mala negociacion entre psoe', 'neg'),
    ('gran final liga barcelona', 'pos'),
    ('erupción galapagos provoca miles muertos', 'neg'),
    ('celtas cortos cantara pais vasco', 'neg'),
    ('cerramos negocio drogas valencia', 'neg')
]
test2 = 'suecia: supermercados reemplazan etiquetas marcas láser (con tecnología )'
print(cl.classify("mala praxis momento exacto episodio hospital"))
print(cl.classify("precio electricidad nuves"))
print(cl.classify("comportamiento barça nefasto ante madrid"))
print(cl.classify("twitter deja miles dolares momento inflación"))
print(cl.classify("historia vaca muere extrañas circumstancias"))
print(cl.classify("mala negociacion entre psoe"))
print(cl.classify("gran final liga barcelona"))
print(cl.classify("erupción galapagos provoca miles muertos"))
print(cl.classify("celtas cortos cantara pais vasco"))
print(cl.classify("cerramos negocio drogas valencia"))

print(cl.accuracy(test))
```

Fig. 11: Codi test machine learning