UNIVERSITAT DE BARCELONA

Treball final de grau

GRAU EN ENGINYERIA INFORMÀTICA

Facultat de Matemàtiques i Informàtica
Universitat de Barcelona

CLASSIFICACIÓ DE SENTIMENTS A TWITTER

Autor: Ricard Planes Vivancos

Director: Dr. Eduardo Urruticoechea
Realitzat a: Departament de llenguatges i sistemes informàtics

Barcelona, June 19, 2021

Abstract

The aim of this project is to implement an unsupervised, fast and low-cost machine learning algorithm capable of identifying the sentiment conveyed by each tweet from a set of tweets in order to calculate the percentage of those positive and negative. To do this, we will also create the Dataset of tweets to classify with which the algorithm will be trained. This project intends to be useful to those companies that during an advertising campaign are looking for a quick first analysis of the Catalan-speaking users' acceptance on Twitter.

Resum

L'objectiu d'aquest projecte és implementar un algoritme de machine learning, no supervisat, ràpid i de baix cost capaç d'identificar el sentiment que transmet cada tuit d'un conjunt de tuits per poder calcular el percentatge de positius i negatius. Per fer-ho, també es crearà el Dataset de tuits a classificar i amb els quals s'entrenarà l'algoritme. Aquest projecte pretén ser d'utilitat a aquelles empreses que durant una campanya publicitaria busquin un ràpid primer anàlisis de l'acceptació que s'hagi tingut a Twitter entre els usuaris catalanoparlants.

Resumen

El objetivo de este proyecto es implementar un algoritmo de machine learning, no supervisado, rapido y de bajo coste capaz de identificar los sentimientos transmitidos por cada tuit de un conjunto de tuits para poder calcular el porcentaje de positivos y negativos. Para ello, también se creará el Dataset de tuits a clasificar y con los cuales se entrenará el algoritmo. Este proyecto pretende ser de utilidad a aquellas empresas que durante una campaña publicitaria busquen un rápido primer análisis de la aceptación que se haya tenido en Twitter entre los usuarios catalanoparlantes.

Agraïments

Vull agrair al meu tutor Eduardo pel seguiment setmanal que ha fet, m'ha ajudat a trobar el camí pel qual volia enfocar el projecte i he pogut plantejar tots aquells dubtes que m'han anat sorgint. També vull agrair a la Montse Nofre, del servei de Tecnologia Lingüística, per la paciència que ha tingut i la gran ajuda que ha significat pel projecte poder obtenir el lema de cada paraula del Dataset. Finalment, agrair als meus pares tant per subvencionar econòmicament tot el grau com per sempre confiar en mi més que jo mateix.

Index

Capítol 1

1 Introducció

1.1 Context

Per comunicar-nos via text a través d'internet utilitzem principalment les xarxes socials i, una de les més populars, Twitter, destaca per la immensa quantitat de gent que opina i debat sobre qualsevol aspecte o tema. En conseqüència, moltes empreses busquen tenir una bona reputació en aquesta xarxa social i es tenen en compte les opinions que es donen tant de l'empresa com d'aquelles novetats que anuncien. Sent així, no és d'estranyar que es facin tants esforços per millorar els algoritmes d'identificació dels sentiments transmesos en el tipus de text informal que ens trobem a Twitter.

Per altra banda, encara que les bases siguin iguals o molt semblants en la majoria de casos, els algoritmes de detecció d'emocions són específics per cada idioma. Per exemple, l'anglès és un idioma on es concentren molts esforços per obtenir resultats al considerar-se l'idioma internacional i el més utilitzat a occident, però, en contra, es treballa molt poc amb les llengües minoritàries com pot ser el català.

1.2 Motivació i objectiu principal

L'objectiu d'aquest projecte és crear un algoritme on, recollint una determinada quantitat de tuits en català, es pugui classificar automàticament quins d'ells transmeten sentiments positius i quins d'ells negatius amb una certa fiabilitat i, per tant, també poder obtenir la quantitat i el percentatge de tuits que transmeten cada sentiment.

La funcionalitat i motivació del projecte és la utilitat que aquesta eina pot tenir per a totes aquelles empreses que presenten productes o anuncien novetats en català per la xarxa social de Twitter, ja que poden recopilar tots els tuits que reaccionen, per exemple, a un determinat anunci i valorar automàticament l'acceptació que ha tingut a trets generals, obtenir percentatges i, si es volgués, analitzar exemples de tuits d'un determinat sentiment per identificar, en cas negatiu els motius de les queixes o, en cas positiu, aclarir el que ha agradat.

1.3 Objectius específics

A partir de l'objectiu principal es poden definir diferents objectius específics:

- Estudiar els diferents tipus d'algoritmes per detectar sentiments automaticament
- Crear un Dataset de tuits en català
- Tractar i netejar els tuits del Dataset
- Implementar i entrenar l'algoritme seleccionat per identificar tuits positius i negatius.
- Implementar una versió de l'algoritme on també valori els tuits neutres
- Crear una interfície d'usuari
- Analitzar els resultats i comparar les dues versions.

1.4 Planificació temporal

Per planificar la feina a fer al llarg de les setmanes es van plantejar unes dates límit en les quals s'havien d'haver assolit certs objectius del projecte, el plantejament és el següent.

Fase del projecte	Data	Tasques
Introducció	8 de març	• Estructura del projecte • Estudiar algoritmes de detecció de sentiments en text
Implementació	5 d'abril	• Obtenir dades de Twitter i tractar-les
	19 d'abril	• Word Embedding
	26 d'abril	• K-Means
	10 de maig	• Tfidf score i prediccions
Anàlisi de Resultats	24 de maig	• Analitzar els resultats obtinguts
Document	14 de juny	• Acabar de redactar el document
	19 de juny	• Revisió del projecte

Figura 1: Taula amb la planificació plantejada

Aquestes dates s'han intentat respectar en la mida del possible, però diferents aspectes amb els quals no es contava en plantejar-lo, per exemple el fet de necessitar replantejar el tractament i neteja del Dataset durant l'anàlisi de resultats, han fet que no sigui del tot possible i s'ha hagut d'adaptar.

1.5 Estructura del document

A continuació hi ha una breu descripció del contingut de cada capítol del projecte:

- **Capítol 1, Introducció:** Introducció al projecte, objectius plantejats, motivació i planificació.
- **Capítol 2, Marc teòric:** Estudi sobre el Processament de llenguatge natural, els diferents mètodes per poder identificar sentiments automàticament i raonament del mètode més indicat a utilitzar en aquest projecte.
- **Capítol 3, Obtenció de dades:** Explicació de com s'han obtingut les dades de Twitter i com s'han tractat, quines decisions s'han pres i la seva motivació.

- **Capítol 4, Algoritme:** Explicació de la idea, estructura i implementació de l'algoritme, explicació del funcionament de les eines utilitzades i funcionament de la interfície del programa.
- **Capítol 5, Resultats:** Anàlisi dels resultats obtinguts, comparació de les dues versions de l'algoritme i raonament dels motius d'aquests resultats.
- **Capítol 6, Conclusions i treball futur:** Conclusió general del projecte i plantejament de possibles futures millores.

Capítol 2

2 Marc teòric

2.1 Processament de llenguatge natural

El processament de llenguatge natural (NLP) és el camp de la informàtica enfocat a estudiar la comprensió del llenguatge humà amb l'objectiu de què els ordinadors siguin capaços d'analitzar-lo, interpretar-lo i donar-li un significat.

El NLP es pot donar a diferents escales sent la més senzilla el nivell de paraula, passant pel nivell de frase, paràgraf i finalment a escala document. Estudiar el llenguatge a escala de paraula és el repte més simple, no obstant això, quan arribem a nivell de frase ens trobem amb moltes dificultats com poden ser les ambigüitats, ironies o fins i tot erros gramaticals que pot arribar a haver-hi en el llenguatge textual.

Les principals funcionalitats del processament de llenguatges naturals són les següents:

- **Resum de textos:** Ser capaç de detectar automàticament quines parts d'un text són més essencials i de quines es pot prescindir per reduir-ne la mida perdent la mínima informació possible.
- **ChatBots:** Poder mantenir converses de forma fluida amb un usuari i resoldre els dubtes que tingui de manera automàtica.
- **Anàlisi de sentiments:** Poder classificar automàticament els sentiments que transmet un text.

- **Classificació de textos:** Ser capaç de detectar el tema o temes tractats en un text i crear categories.
- **Traducció:** Poder traduir automàticament un text d'un idioma a un altre.

2.2 Anàlisi de sentiments

L'anàlisi de sentiments o mineria d'opinió és l'aplicació del processament de llenguatge natural amb l'objectiu de determinar automàticament els sentiments que transmet una frase, text o document. Els algoritmes d'anàlisi de sentiments solen identificar la polarització del text, és a dir, si transmet un sentiment positiu, negatiu o neutre, però hi ha variacions que pretenen identificar el sentiment amb més precisió, per exemple, detectar si es transmet fúria, tristesa, alegria o desesperació.

Independentment d'aquesta classificació, hi ha 5 mètodes principals de detecció de sentiments en textos, el basat en paraules clau, el basat en normes, el basat en el machine learning, el basat en deep learning i l'híbrid entre aquests dos últims.

2.3 Metodes d'anàlisi de sentiments

2.3.1 Basat en paraules clau

El mètode d'anàlisi de sentiments basat en paraules és extremadament simple, es basa en analitzar el text a la recerca de paraules a les quals se les hi hagi assignat prèviament una emoció en concret, anomenades paraules clau, i concloure el sentiment general del text a partir d'aquestes.

Aquest mètode és massa propens a fallar per diferents inconvenients, és un mètode que només funciona quan s'expressa el sentiment explícitament amb paraules a les quals prèviament se les hi ha assignat una emoció. A més, és poc pràctic adaptar-lo a interpretar correctament les negacions, que inverteixen la polaritat de la frase i, per tant, el significat de les paraules clau, causant així errors de predicció.

2.3.2 Basat en normes

En el mètode basat en normes es creen manualment un conjunt de normes a partir de conceptes lingüístics per després aplicar-les als textos corresponents i extreure el sentiment que transmeten.

Es tracta d'un mètode capaç d'interpretar només aquells sentiments expressats de tal manera que almenys una de les normes especificades pugui detectar-lo. També s'ha comprovat que aquest mètode, al dependre completament de normes basades en conceptes lingüístics, és molt sensible a la qualitat del text, si el text no té una gran correcció ortogràfica i gramatical l'eficiència d'aquest mètode es redueix en gran mesura.

2.3.3 Basat en el machine learning

Els mètodes basats en machine learning són tots aquells algoritmes que s'entrenen a partir d'unes dades proporcionades, dades d'entrenament, per aprendre a ser capaços d'identificar sentiments. Hi ha un ampli ventall de tècniques diferents que apliquen aquest mètode.

Hi ha dos grans grups d'algoritmes d'anàlisi de sentiments basats en el machine learning:

- **Supervisats:** Són aquells algoritmes que utilitzen dades a les quals se les hi ha assignat prèviament el sentiment que transmeten i utilitzen aquesta informació per entrenar el model. Aquests sentiments assignats es coneixen com a etiquetes.
- **No supervisat:** Són aquells algoritmes que utilitzen dades sense etiquetar. Per poder classificar sentiments solament amb l'ajuda dels textos del Dataset, l'algoritme ha de ser capaç d'identificar tot allò que caracteritza cada un dels sentiments i aquelles que el diferencien de la resta.

El fet de tenir les dades etiquetades és un avantatge molt important i, per tant, els models supervisats solen obtenir resultats més fiables que els no supervisats. D'afegit, el fet de tenir les dades etiquetades també se sol utilitzar per valorar l'eficiència de l'algoritme al poder comparar els resultats obtinguts amb els assignats

prèviament de forma manual. Per altra banda, etiquetar manualment un conjunt de dades suficientment gran per a poder entrenar un algoritme de machine learning és una feina que implica una quantitat de temps tan gran que en molts casos arriba a ser inviable.

2.3.4 Deep learning

El deep learning és una branca del machine learning que busca aprendre per si mateix a partir del reconeixement de patrons, per aconseguir-ho s'utilitza un tipus d'algoritme anomenat xarxes neuronals artificials. Deep learning destaca per ser una tècnica capaç de resoldre una gran varietat de tasques i d'una alta complexitat de forma molt eficient.

Els principals avantatges d'utilitzar metodologia deep learning per analitzar sentiments en textos és l'alta qualitat dels resultats i l'adaptabilitat a diferents circumstàncies que ofereix l'algoritme. En contra, el deep learning requereix d'una quantitat tan gran de dades etiquetades per entrenar el model que en molts casos no és assequible.

2.3.5 Híbrid

Hi ha una gran varietat d'algoritmes que entren en la categoria d'híbrids, són tots aquells que adquireixen tant algunes característiques de machine learning com d'altres de deep learning.

Aquest tipus d'algoritme busca millorar els resultats intentant obtenir alhora els avantatges que ofereixen cada un dels dos mètodes, machine learning i deep learning. Per altra banda, el desavantatge del model híbrid és que, en aplicar les característiques d'un mètode, hi ha el risc d'heretar també alguns desavantatges d'aquest.

2.4 Decisió de l'algoritme

El mètode basat en paraules clau és un mètode poc recomanat al ser massa poc fiable i només identificar sentiments expressats explícitament. Altrament, el

mètode basat en normes també es pot descartar al ser un mètode que pateix un decreixement extrem de l'efectivitat quan les dades no compleixen uns mínims de correcció ortogràfica i gramatical, una característica no compatible amb l'objectiu del projecte, on es busca treballar amb contingut de Twitter, textos extremadament informals.

Es pot concloure que serà necessari utilitzar un algoritme d'aprenentatge automàtic, per tant, abans de valorar quin algoritme és l'adequat, cal valorar les dades de les quals es disposarà per entrenar-lo.

L'objectiu del projecte és agafar un conjunt de tuits i classificar-los, totes les dades de les quals es disposaran serà el contingut dels tuits a classificar, en aquestes condicions hi ha dos possibles plantejaments. El primer d'aquests és utilitzar un Dataset etiquetat de tuits en català suficientment semblants al tipus de tuits que es voldrà classificar per entrenar un algoritme supervisat. El segon plantejament és entrenar l'algoritme amb les mateixes dades a classificar i, per tant, implementar un algoritme no supervisat.

Com no és possible trobar un Dataset en català etiquetat prou idoni per a entrenar satisfactòriament un algoritme supervisat i tampoc es disposa dels recursos per crear-lo, serà necessari implementar un algoritme de machine learning no supervisat.

Capítol 3

3 Dataset

El Dataset és el conjunt de tuits a classificar que alhora s'utilitzaran per entrenar l'algoritme. Les decisions d'obtenció de les dades i el tractament d'aquestes és decisiu, determinen en gran manera els resultats i, per tant, l'èxit del projecte.

3.1 Obtenció de les dades

Per seleccionar els tuits, s'ha utilitzat l'únic criteri de què aquests estiguin en català, el Dataset estarà format per tuits sense filtratge de temàtica, ideologia, data de publicació o localitat. Les eines a utilitzar per a la tasca d'extreure tuits en català seran Tweepy i Texblob.

3.1.1 Tweepy

Tweepy és una llibreria de Python que facilita l'accés a la API de Twitter per així poder extreure tuits automàticament. Aquesta llibreria ofereix totes aquelles funcionalitats disponibles a l'API de Twitter, entre elles retornar una quantitat de tuits que compleixin unes condicions en concret.

3.1.2 TextBlob

TextBlob és una altra llibreria de Python per processar textos, aquesta llibreria es pot utilitzar per detectar l'idioma d'un text. El filtratge d'idioma que ofereix Tweepy és massa poc fiable, TextBlob s'utilitzarà com a segon filtratge d'idioma.

TextBlob no permet fer crides il·limitades a la llibreria, per no superar el límit de crides s'ha de fer una pausa de 0,5 segons després de cada crida. Tot i que és una inversió de temps necessària no deixa de ser un gran increment del temps dedicat a l'obtenció de tuits.

3.1.3 Implementació

Utilitzant les funcionalitats de Tweepy i TextBlob es crea un mètode que retorni una taula Dataframe amb el contingut i nombre de caràcters d'una quantitat de tuits passada per paràmetre, en aquest metode és on s'aplica TextBlob per assegurar que tot el contingut del Dataset és en català.

Com utilitzar Tweepy i TextBlob implica que l'obtenció de tuits tardi una quantitat considerable de temps, no és recomanable executar-lo perquè retorni una gran quantitat de tuits. Per obtenir tots els tuits necessaris s'utilitza un bucle iteratiu que recopila tuits de 100 en 100, aquest bucle té la següent estructura.

- S'obté una taula dataframe amb 100 tuits.
- S'obre l'arxiu .csv amb un altre dataframe on estan guardats tots els tuits obtinguts.
- Es sumen els dos dataframes.
- S'eliminen duplicats de la suma de dataframes.
- Es sobreescriu al fitxer .csv el dataframe obtingut.

Aproximadament, cada una d'aquestes iteracions tarda 50 segons i afegeix 80 tuits de mitja al Dataset, ja que se sol descartar un 20% com a duplicats.

3.2 Tractament de les dades

El tractament del Dataset és un procés igual d'important que obtenir de dades en si. Es tracta del procés de preparar el text perquè l'entrenament de l'algoritme sigui òptim. Aquest procés principalment es centra en eliminar aquella informació innecessària o que pot confondre l'algoritme i adaptar aquella informació que sí que és rellevant. Tot i que la preparació de les dades pot tenir una base comuna per la

gran majoria de Datasets i algoritmes, és un procés que sempre s'ha d'adaptar a cada situació plantejada. Per decidir com tractar el Dataset s'ha tingut en compte principalment el fet que les dades a tractar són tuits i quins objectius es volen assolir amb l'algoritme.

3.2.1 Delimitar tamany:

Perquè sigui possible analitzar les paraules que envolten cada paraula i obtenir conclusions satisfactòries, cal que aquests tuits tinguin un cert nombre de paraules. Si un tuit conté menys de 8 paraules es considera que no és vàlid per fer aquesta anàlisi i es descarta del Dataset.

3.2.2 Substituir majúscules

Encara que una mateixa paraula amb majúscules o minúscules té el mateix significat, dues paraules iguals on només canvien certes lletres en majúscules solen ser identificades com a paraules completament diferents, per evitar això es processa el Dataset substituint totes les lletres majúscules per la seva versió en minúscules.

3.2.3 Substituir mencions

Un tret molt característic dels tuits és que són molt propensos a contenir citacions, mencions a altres comptes de la xarxa. Sense adaptar el Dataset, l'algoritme consideraria que cada una d'aquestes citacions és una paraula. Tot i que per evitar-ho una opció pot ser eliminar totes aquestes cites, és més encertat detectar totes les citacions i substituir-les per un conjunt de caràcters fàcilment diferenciable que l'algoritme tracti com a una única paraula amb la funció de representar tots aquells noms propis que es tractaran com a anònims, ja que no interessa a qui mencionen. Aquest conjunt de caràcters és "nompropi", ja que és molt poc probable que aparegui com a contingut d'un tuit de cap altra manera. Per identificar les citacions es busquen totes aquelles paraules que comencin amb el caràcter "@".

3.2.4 Separar signes de puntuació

Els signes de puntuació són un altre tret a tenir en compte quan es tracta d'identificar sentiments, el significat d'una frase canvia significativament si al final

hi ha un signe d'interrogació o un signe d'exclamació. Per tractar com a paraules independents els signes de puntuació s'afegirà un espai de separació entre el signe i la paraula a la qual està enganxat.

3.2.5 Eliminar enllaços

Com s'està treballant amb contingut de Twitter s'ha de tenir en compte la possibilitat que alguns tuits continguin un o més enllaços, una informació que tampoc interessa analitzar, ja que no aporta informació sobre el sentiment que es transmet. Sent així, totes aquelles paraules que comencin amb "https seran eliminades del Dataset.

3.2.6 Detecció d'expressions

És important la detecció d'expressions o frases fetes comunes al Dataset. Per detectar-ho automàticament s'utilitza la llibreria Phrases, que busca conjunts de poques paraules que se solen trobar juntes amb una certa freqüència al llarg del Dataset.

3.2.7 Obtenir forma base de cada paraula

Preparar les dades per facilitar a l'algoritme la identificació de la mateixa paraula al llarg del Dataset encara que estigui escrita de diferent manera és vital per optimitzar l'entrenament. Per altra banda, la immensa majoria de paraules en Català, com en molts altres idiomes, tenen la característica de tenir diferents formes, ja sigui pel gènere, el temps verbal en el cas dels verbs, si són plurals o singulars o per una gran varietat de motius. Aquesta característica de les paraules impossibilita que l'algoritme pugui identificar la mateixa paraula si està expressada de diverses maneres. Per evitar aquesta pèrdua d'informació s'ha de substituir cada paraula del Dataset per la seva forma base.

Obtenir la forma base de cada paraula del Dataset requereix d'una eina externa molt concreta de la que, en català, no existeix cap pública. El servei de Tecnologia Lingüistica de la UB ofereix la possibilitat de processar textos per un lematitzador del qual disposen anomenat Freeling. Processant el Dataset s'obté una

versió d'aquest on totes les paraules han sigut substituïdes per la seva versió lema.[1]

3.2.8 Elminar emoticones gràfiques

Les emoticones són símbols que solen expressar explícitament el sentiment que es vol transmetre. Tot i que poden arribar a ser de gran ajuda en molts casos per concloure el sentiment d'un tuit, no solen tenir un comportament estable en els algoritmes de processament de llenguatge natural al no seguir les estructures del text i, per tant, no complir en la majoria de casos les hipòtesis en les quals es basa l'algoritme. També, l'eliminació de les emoticones gràfiques és necessària per processar el text pel lematitzador.

3.2.9 Dividir els tuits

Com a últim pas del tractament de les dades cal separar per paraules cada tuit. Un tuit, en lloc de ser un string, serà una llista de strings on cada string sigui una paraula del tuit.

Procés	Resultat
Tuit original	"@tv3cat Quin dia creieu que farà avui? 😕 https://ccma.cat/tv3"
Substituir majúscules	"@tv3cat quin dia creieu que farà avui? 😕 https://ccma.cat/tv3"
Substituir mencions	"nompropi quin dia creieu que farà avui? 😕 https://ccma.cat/tv3"
Separar símbols de puntuació	"nompropi quin dia creieu que farà avui ? 😕 https://ccma.cat/tv3"
Eliminar enllaços	"nompropi quin dia creieu que farà avui ? 😕"
Eliminar emoticones	"nompropi quin dia creieu que farà avui ?"
Forma base	"nompropi quin dia creure que fa avui ?"
Dividir per paraules	["nompropi", "quin", "dia", "creure", "que", "fa", "avui", "?"]

Figura 2: Exemplificació de cada pas del procés del tractament d'un tuit.

[1]Lema: Forma base d'una paraula, com la trobaríem a un diccionari, per exemple el lema d'un verb és l'infinitiu i per un substantiu la forma en singular.

Capítol 4

4 Algoritme

4.1 Hipòtesi

La suposició en la qual es basa l'algoritme és que aquelles paraules que transmeten sentiments positius i negatius estan envoltades per paraules similars. Com més semblant sigui el significat de dues paraules, més semblants seran aquelles paraules que les envolten. L'algoritme analitzarà les similituds mitjançant aquesta suposició per classificar les paraules de cada tuit entre positives, negatives i, si cal, neutres.

Per exemplificar aquesta hipòtesi, si al Dataset es troben les paraules "encantador", "amable" i "desagradable", tot i que les paraules "encantador" i "amable" no tinguin el mateix significat, sí que ambdues transmeten sentiments positius mentre que "desagradable" en transmet de negatius. Per tant, les paraules al voltant d'"amable" i "desagradable" seran molt més semblants entre elles que si es comparen amb les paraules que envoltin "desagradable".

4.2 Estructura

L'algoritme es divideix principalment en 5 fases:

Word Embedding:
El Word Embedding consisteix en representar les paraules del Dataset com a vectors de números. En el cas d'aquest projecte serà durant el Word Embedding quan es buscaran les similituds i relacions entre les paraules del Dataset, a cada paraula se li assignarà un vector anomenat vector paraula que representi aquestes similituds. Per aquesta fase s'utilitzarà l'algoritme de Word2vec, una eina ideal per aquest tre-

ball.

Classificació de les paraules:
S'utilitzen els vectors extrets amb Word2vec per classificar les paraules en tants grups com sentiments es vulguin identificar. Inicialment es classificaran els vectors en 2 grups per dividir les paraules entre positives i negatives, per la versió on es té en compte la opció de que hi hagi tuits neutres s'afegirà un tercer grup. Per classificar els vectors s'utilitzarà el mètode d'agrupament K-means.

Càlcul del valor sentiment:
El valor sentiment és el valor numeric que indica quantitativament el sentiment que transmet una paraula. Es calcula a partir de la distància entre el vector paraula i el centre del grup de paraules al qual pertany.

Càlcul de la rellevancia:
En aquesta fase es busca quantificar numèricament la importància de cada paraula del Dataset tenint en compte la seva freqüència d'aparició.

Predicció final
L'algoritme conclou el sentiment del tuit utilitzant el valor sentiment i la rellevància de cada paraula.

4.3 Word embedding

Word Embedding és la pràctica de representar paraules com a vectors de números. Es tracta d'un procés necessari en la immensa majoria d'algoritmes NLP, ja que treballar amb números que tinguin paraules assignades és molt més còmode i eficient que no pas treballar amb cadenes de caràcters com és el text.

El vector assignat a cada paraula simula un punt a un espai d'un nombre de dimensions concret, depenent de les proximitats que es trobin entre tots aquests vectors de l'espai serà possible classificar aquests vectors en diferents grups i, per tant, les paraules que representen.

Els valors concrets de cada vector i les proximitats a altres vectors serà assignat a partir de la suposició inicial on es busquen similituds entre paraules analitzant

aquelles paraules que les envolten. Com més semblants siguin els conjunts de paraules que envolten dues paraules, més propers es trobaran els seus corresponents vectors a l'espai.

4.3.1 Word2vec

Word2vec és una xarxa neuronal que processa textos convertint les paraules en vectors. Donat un text o conjunt de textos, Word2vec crea un diccionari amb totes les paraules que apareixen i retorna un conjunt de vectors on cada un d'ells representa una paraula d'aquest diccionari.

Mitjançant un entrenament, Word2vec extreu les semblances de les diferents paraules a partir de la suposició inicial, analitzant aquelles paraules que envolta cada paraula. Per fer-ho necessita una variable numèrica anomenada "window" o finestra que indica a quina distancia ha d'estar una paraula per considerar-se propera. Si aquest valor és 2, les paraules properes seran les dues anteriors i les dues posteriors a la paraula en qüestió.

Per analitzar la semblança entre paraules, Word2vec emparella cada paraula amb cada una de les seves paraules properes. La semblança entre paraules s'extraurà a partir de la quantitat de cops que es repeteix cada una de les parelles de paraules.

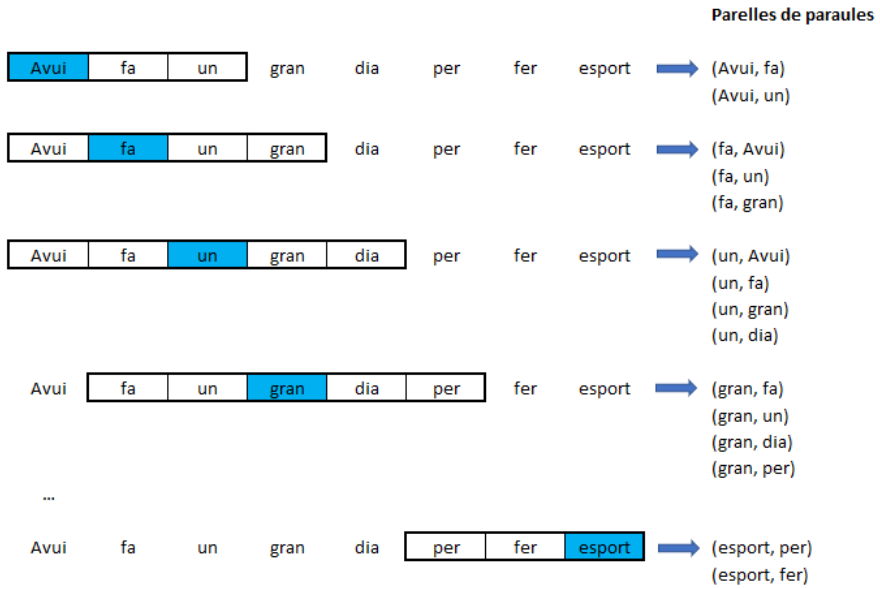

Figura 3: Exemple de l'emparellament de paraules amb una finestra de mida 2.

Implementació del Word2vec:
La versió de l'algoritme Word2vec que s'utilitza en el projecte és la versió que ofereix la llibreria de codi obert Gensim amb la que s'obtindran uns vectors paraula de dimensió 100.

Primerament es crea un objecte Word2Vec, al qual se li assigna pels paràmetres del constructor de l'objecte aquelles característiques que vulguem concretar per adaptar l'entrenament a les dades que seran entrenades, un cop s'ha inicialitzat l'objecte Word2Vec, s'utilitzen dos mètodes propis de d'aquest objecte.

El primer d'aquests mètodes identifica totes les paraules que apareixen al Dataset per crear un diccionari on a cada una d'aquestes paraules se li assigna un índex. El segon mètode és el que entrena l'algoritme per obtenir els vectors corresponents a cada paraula, a aquest mètode també se li ha de passar per paràmetre el Dataset.

A continuació s'expliquen quins paràmetres es modifiquen respecte als predefinits, perquè serveixen i el raonament darrere del valor assignat a cada un d'ells.

Window:
És el paràmetre que determina el concepte de proximitat a les paraules, el valor numèric d'aquest paràmetre indica en quin rang ha d'estar una paraula per considerar-se pròxima. És un valor que s'ha d'adaptar a la mida de les frases, al treballar amb tuits de mínim 8 paraules se li ha assignat un valor de 4. Word2vec emparellarà cada paraula amb les 4 anteriors i les 4 posteriors.

Min_count:
Aquest paràmetre representa amb un valor enter la freqüència total mínima que ha de tenir una paraula per no ser ignorada. La freqüència total és simplement la quantitat de cops que apareix un valor en tot el Dataset. Les paraules que apareixen menys cops que el valor que tingui "min_count" no es tindran en compte.

Tenir aquest paràmetre a un valor superior a 0 serveix molt eficientment per filtrar totes aquelles paraules mal escrites per error, ja que és prou difícil que una paraula s'escrigui malament de la mateixa manera més de 10 cops. A part, també serveix pel motiu més intuïtiu, no tenir en compte les paraules més irrellevants. Tot i que el paràmetre per defecte és 5, s'ha incrementat a 10 perquè, al ser un paràmetre que només té en consideració la quantitat de cops que apareix una paraula, s'ha d'adaptar a la quantitat de text amb la que s'està tractant. Treballar

amb una gran quantitat de tuits augmenta les possibilitats de què es repeteixi un error 5 cops i per tant el filtratge no sigui tan eficaç.

Sample:
El paràmetre "sample" és un valor que serveix com a llindar per determinar a partir de quina freqüència s'han de reduir aleatòriament la freqüència de mostreig perque no hi hagi paraules amb un mostreig massa alt. El valor per defecte d'aquest paràmetre és 0.001 tot i que la documentació de Gensin indica que el rang d'utilitat d'aquest paràmetre és entre 0 i 1e-5[2].

Negative Sampling:
Durant el procés d'entrenament, Word2Vec utilitza la funció SoftMax, una funció molt costosa, especialment quan s'ha d'aplicar per a molts valors. Word2vec ofereix una alternativa coneguda com a negative sampling, consisteix en calcular una aproximació significativament menys costosa del càlcul que fa SoftMax convertint el problema de classificació multinominal [3] que ha de calcular SoftMax en un problema de classificació binària.

Aquesta classificació binària consisteix en barrejar una parella de paraules real que consti de la paraula central i una paraula de la seva proximitat entre diferents parelles falses. Aquestes parelles falses estan formades per la paraula central i una paraula que serà seleccionada d'un grup de paraules conegut com a mostreig negatiu. Durant l'entrenament que es necessita perquè l'algoritme sigui capaç d'identificar la parella real de les falses s'aniran formant els vectors paraula. Word2vec modifica una mica aquest algoritme de classificació perquè la probabilitat d'una paraula per ser seleccionada com un mostreig negatiu estigui relacionat amb la seva freqüència, sent les paraules més freqüents més probables de ser seleccionades com a mostreig negatiu.

El paràmetre amb el qual s'indica si utilitzar Negative Sampling i les condicions en les quals utilitzar-lo té el nom de "negative", si aquest paràmetre té assignat un valor major de 0 s'aplica el negative sampling, en cas contrari no. El valor del paràmetre especifica quantes paraules soroll, les paraules que formaran part del mostreig negatiu, hi haurà. Per un volum de dades com el del Dataset que es vol utilitzar es recomana utilitzar un valor d'entre 5 i 20[4].

[2] `https://radimrehurek.com/gensim/models/word2vec.html`
[3] Classificació entre més de dos grups.
[4] `https://medium.com/@makcedward/how-negative-sampling-work-on-word2vec-7bf8d545b116`

4.4 Classificació de les paraules

La classificació de paraules entre positives i negatives s'obté analitzant la posició a l'espai de tots els vectors paraula, buscant agrupar aquelles més properes. Per agrupar les paraules per proximitat s'utilitzarà l'algoritme K-means, concretament la implementació de Scikit-learn.

L'algoritme K-means primer col·loca punts aleatòriament a l'espai, se'n col·loquen tants com grups diferents es vulguin diferenciar, on cada punt representa la ubicació del centre de cada grup. L'algoritme buscarà la posició ideal per cada un dels punts centrals a partir dels vectors que hi hagi a l'espai, per fer-ho s'executarà iterativament un càlcul que apropi a cada punt a la posició més encertada com a centre de grup. Quan després de diverses iteracions cap dels punts centrals canviï la seva ubicació es considerarà que s'ha trobat el centre de cada grup.

Un cop s'obtenen els grups de paraules s'ha d'indicar manualment quin sentiment representa cada un. La decisió es pren a partir de l'anàlisi d'aquelles paraules més properes al centre de cada grup. En el cas d'aquest projecte, el conjunt que s'ha considerat negatiu conté paraules com "narcisista", "malvat" o "humiliant". Per altra banda el conjunt de paraules positives en té d'altres com "projecte_guanyador", "esportiva" o "escolars". Es pot apreciar com les paraules positives són molt menys polaritzades que no pas les negatives, que tenen una alta connotació negativa, és per això que en aquest punt ja es pot deduir que les paraules negatives s'agrupen més fàcilment que les positives i, per tant, l'algoritme pot donar millors resultats a l'identificar els tuits negatius.

4.4.1 Identificació de paraules neutres:

Per adaptar l'algoritme a la versió on també es classifiquen les paraules neutres cal aplicar l'algoritme de K-means per dividir les paraules en 3 grups.

L'assignació manual del sentiment a cada grup de paraules a partir de les més properes al centre es complica en aquesta versió. Tot i que el grup de paraules negatives se segueix diferenciant amb facilitat, no es pot identificar el grup de paraules positives i el de neutres amb certesa. Per solucionar aquesta problemàtica, un

cop l'algoritme hagi sigut completament implementat s'haurà de comprovar quina assignació ofereix millors resultats.

4.5 Càlcul del valor sentiment

El valor sentiment és un valor numèric que indica com de polaritzada està la paraula mentre el signe del número determinarà a quin dels grups pertany.

A partir de la suposició de què com més propera al centre del seu grup estigui una paraula més intensa serà representant aquell sentiment, el valor numèric de sentiment es calcularà a partir de la distància que hi hagi entre cada paraula i el centre del seu grup. Per obtenir la distància de cada vector al centre del seu corresponent grup, es calcularà la distància a tots els centres i es guardarà només la distancia més curta. Un cop es té la distància, per obtenir un valor sentiment inversament proporcional a la distancia que hi hagi entre el centre i el vector, el valor sentiment es calcularà fent la inversa de la distància mínima, és a dir, el valor sentiment d'una paraula serà igual a 1 dividit entre la distància al centre més proper.

Valor sentiment = 1/distancia minima

Per poder identificar amb aquest valor si la paraula forma part del grup de positius o negatius se li assignarà a aquest valor sentiment el signe corresponent. En cas que una paraula formi part del grup de paraules positives, el seu valor sentiment serà positiu, altrament, si la paraula forma part del grup de paraules negatives, el seu valor sentiment serà negatiu.

4.5.1 Valor sentiment de les paraules neutres

El càlcul del valor sentiment tenint en compte les paraules neutres es basa en la següent suposició. Com més allunyat de 0 estigui el valor sentiment d'una paraula, més intensament reflectirà el sentiment del grup al qual pertany aquella paraula. Tenint això en consideració, aquelles paraules considerades neutres, que no representen cap sentiment, han de tenir un valor sentiment de 0.

4.6 Importància de les paraules

Amb l'objectiu de calcular el sentiment que transmet un tuit sencer, cal calcular per cada paraula un valor numèric de rellevància i així identificar aquelles paraules de cada tuit a les quals cal donar més pes al seu corresponent valor sentiment.

4.6.1 Formula Tf-idf:

Tf-idf (Term Frequency-Iverse Document Frequency) és una fórmula amb la qual calcular com és de rellevant una paraula en una frase que forma part d'un grup de frases. Per calcular la rellevància d'una paraula, aquesta fórmula utilitza la freqüència amb la qual apareix la paraula a la frase i la freqüència inversa de la paraula en tot el conjunt de frases.

Freqüència d'un terme (Tf):
Aquest valor indica com de freqüentment apareix una paraula en una frase, tuits en el cas d'aquest projecte. Es calcula dividint el nombre de cops que apareix aquella paraula entre el nombre total de paraules de la frase. Sent x una paraula la fórmula és la següent.

Tf(x) = cops que apareix x a la frase / nombre de paraules de la frase

Freqüència Inversa de Document(Idf):
La freqüència inversa indica en quantes frases apareix una paraula, es calcula dividint el nombre de frases totals entre el nombre de frases on apareix la paraula en qüestió. Aquest valor no té en compte si la paraula apareix més d'un cop en una frase, només es busca saber en quants documents apareix. Sent x una paraula la fórmula és la següent.

Idf(x) = nombre de frases/nombre de frases on apareix x

La rellevància d'una paraula es calcula multiplicant la freqüència d'aquesta per la freqüencia inversa de document, com més alt sigui el valor obtingut més importància tindrà la paraula en el tuit.

Per obtenir els valor Tf-idf s'utilitza TfidfVectorizer, també de la llibreria Scikit-learn, qué ofereix funcionalitats per obtenir un diccionari on cada paraula del Dataset tingui assignat un valor Tf-idf.

Amb tfidfVectorizer s'obté una llista de tants elements com paraules totals hi ha al Dataset. Cada element d'aquesta llista està formada per una tupla, que conté dos enters, i un número decimal. El primer enter de la tupla indica l'índex de la frase en la qual es troba la paraula mentre que el segon enter indica l'índex al diccionari de la paraula de la qual es tracta. El número decimal indica la importància de la paraula en qüestió.

Amb el diccionari amb els valors de rellevància creat, es fa una còpia del Dataset en la qual se substitueix cada paraula pel seu valor Tf-idf. La versió de l'algoritme on s'identifiquen també les paraules neutres no rep cap modificació respecte a l'original durant el càlcul de la rellevància de les paraules.

4.7 Predicció final

Per obtenir la predicció final del sentiment que transmet un tuit es calcula un valor numèric de sentiment a partir dels valors de sentiment i de rellevància de cada paraula que conté. S'obtenen 2 vectors per cada tuit, un amb els valors sentiment de cada paraula i l'altre amb els valors de rellevància, també de les paraules de la frase.

Per obtenir un valor numèric que indiqui el sentiment que transmet un tuit, es fa el producte escalar d'aquests dos vectors, sumant així la multiplicació del valor sentiment i la rellevància de cada paraula de la frase.

		Avui	fa	un	bon	dia	per	fer	esport	
Vector Tf-idf	[	6	4	1	10	9	3	5	7	]
Vector sentiment	[	5	-3	2	15	8	-2	4	7	]

Ràtio de sentiment del tuit = 6 * 5 + 4 * -3 + 1 * 2 + 10 * 15 + 9 * 8 + 3 * -2 + 5 * 4 + 7 * 7

Figura 4: Exemple de com es calcula el producte escalar, tant el tuit com els valors dels vectors són inventats per fer més clar l'exemple.

Un cop s'obté el valor numèric de sentiment de cada tuit es comprovar el signe d'aquest valor per concloure el sentiment que transmet el tuit. Si el valor de sentiment del tuit és un nombre positiu, el tuit serà classificat com a tuit que transmet sentiments positius, en cas contrari, el tuit serà classificat com a tuit que transmet sentiments negatius.

4.7.1 Identificació de tuits neutres:

Com succeïa amb el valor sentiment de les paraules, com més alt sigui el valor absolut del valor de sentiment d'un tuit, més polaritzat serà el tuit. Quan se sumen nombres positius i negatius s'està obtenint la diferència entre la suma dels nombres positius i la suma dels nombres negatius. En aquest cas, com més alt sigui el valor absolut del valor numèric del tuit, de més haurà destacat un dels sentiments i, per tant, més intens serà el sentiment que transmet el tuit.

Amb aquest raonament, la millor interacció que poden tenir els tuits neutres en aquesta disputa entre valors sentiment positius i negatius és no intervenir, si una paraula és considerada neutre, aquesta sumarà 0 al valor de sentiment del tuit. Com el valor sentiment de les paraules neutres ja és 0, fent el producte escalar, on es multiplica el producte escalar i el valor sentiment, el valor ja es quedarà a 0, no cal modificar la implementació.

Per classificar els tuits a partir del valor de sentiment del tuit també s'ha d'adaptar la implementació. Per valorar si un tuit és neutre s'assignarà un interval on es consideri que els tuits són neutres, si el valor de sentiment del tuit té un valor absolut més petit que l'interval neutre, és a dir, la suma de paraules positives i negatives està massa igualada o la majoria de paraules són neutres, el tuit es considerarà neutre.

4.8 Interfície d'usuari

La interfície del programa és on s'etiqueten manualment els tuits Es mostra un tuit no etiquetat del Dataset, es selecciona manualment el sentiment que transmet mitjançant botons i un cop s'ha seleccionat el sentiment es mostra la predicció feta per cada versió de l'algoritme. A l'apretar el botó de "Següent tuit" es mostra el següent tuit a etiquetar i s'actualitzen els diferents percentatges.

La interfície mostra, per les dues versiones de l'algoritme, tant la predicció del percentatge de tuits de cada sentiment com el percentatge d'encert total i individual per cada sentiment. El percentatge de cada sentiment es mostra pel total del Dataset i per la mostra etiquetada.

Aquesta interfície, implementada amb la llibreria tkinter, utilitza les taules que s'obtenen en els fitxers .csv per fer els càlculs de percentatges. Per guardar els tuits ja etiquetats s'obra un altre fitxer .csv amb una taula d'una sola columna on hi ha totes les assignacions de sentiment manuals, aquest fitxer s'actualitza amb les noves etiquetes quan es tanca el programa.

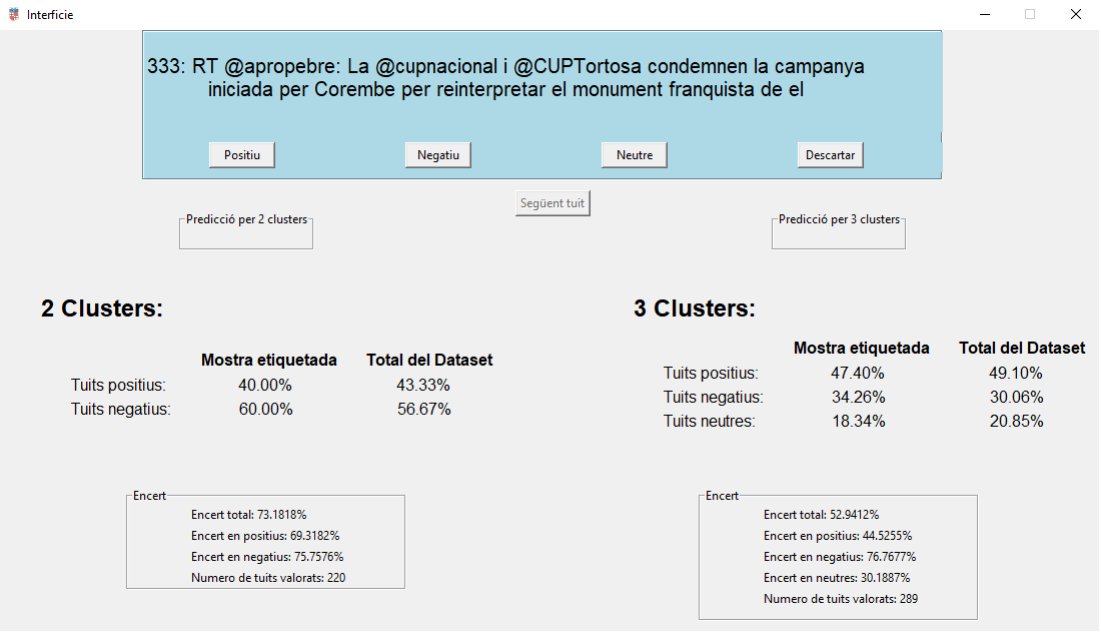

Figura 5: Interfície d'usuari del projecte

Capítol 5

5 Resultats

Per valorar l'encert de l'algoritme ha calgut assignar manualment un sentiment a una fracció del Dataset i comparar aquests sentiments assignats amb les prediccions de l'algoritme. Per la versió de 2 clusters s'han etiquetat manualment 220 tuits interpretats com a positius o negatius. En canvi, per la versió de 3 clusters s'han etiquetat 289 tuits com a positius, negatius o neutres.

L'estudi dels resultats de l'algoritme de classificació de sentiments en tuits es realitza mitjançant matrius de confusió, taules comparatives i gràfiques. Cal estudiar el percentatge d'encert que ha obtingut l'algoritme i la ràtio de tuits assignats a cada sentiment. La fase de resultats també inclou la comparació entre la versió original de l'algoritme, la de 2 clusters, i la que té en consideració la possibilitat de trobar tuits neutres, 3 clusters.

5.1 Encert de l'algoritme

5.1.1 Matriu de confusió

La matriu de confusió és una eina utilitzada en els problemes de classificació supervisats, permet veure el rendiment que ha tingut l'algoritme a analitzar. Tot i que es treballa amb un algoritme no supervisat es pot aplicar aquesta matriu amb els tuits etiquetats manualment.

Les columnes de la matriu seran les prediccions obtingudes per l'algoritme mentre que les files els valors reals. El valor de cada cel·la és la suma de tots els tuits que han sigut predits com el sentiment de la columna i que el seu valor real és l'indicat

per la fila.

Per assegurar que els resultats són correctes es pot sumar cada columna, la suma de cada columna ha de ser el nombre de tuits predits amb aquell sentiment, la suma de les sumes de totes les columnes ha de ser el nombre total de tuits. Amb les files es pot seguir el mateix procediment, si els valors de la matriu són correctes la suma de sumes de les files també ha de ser el nombre total de tuits.

		Prediccions		
		Positiu	**Negatiu**	**Total/Suma**
Reals	**Positiu**	Positius bons	Falsos negatius	Suma de positius reals
	Negatiu	Falsos positius	Negatius bons	Suma de negatius reals
	Total/Suma	Suma de positius predits	Suma de negatius predits	Suma total de tuits

Figura 6: Explicació de la matriu de confusió per 2 clusters.

		Prediccions		
		Positiu	**Negatiu**	**Total/Suma**
Reals	**Positiu**	61	32	93
	Negatiu	27	100	127
	Total/Suma	88	132	**220**

Figura 7: Matriu de confusió per 2 clusters

La matriu de confusió de dos clusters mostra com la quantitat de prediccions encertades tant de tuits positius com negatius és significativament superior a les prediccions errònies.

		Prediccions			
		Positiu	**Negatiu**	**Neutre**	**Total/Suma**
Reals	**Positiu**	Positius correctes	Negatius que haurien de ser positius	Neutres que haurien de ser positius	Suma de positius reals
	Negatiu	Positius que haurien de ser negatius	Negatius correctes	Neutres que haurien de ser negatius	Suma de negatius reals
	Neutre	Positius que haurien de ser neutres	Negatiusque haurien de ser neutres	Neutres correctes	Suma de neutres reals
	Total/Suma	Suma de positius predits	Suma de negatius predits	Suma de neutres predits	Suma total de tuits

Figura 8: Explicació de la matriu de confusió per 3 clusters.

		Prediccions			
		Positiu	**Negatiu**	**Neutre**	**Total/Suma**
Reals	**Positiu**	61	15	17	93
	Negatiu	31	76	20	127
	Neutre	45	8	16	69
	Total/Suma	137	99	60	**289**

Figura 9: Matriu de confusió per 3 clusters

Amb la matriu de confusió de 3 clusters s'observa que la quantitat de tuits on la predicció coincideix amb la realitat és significativament superior que els altres dos casos. Al dividir en 3 grups també es pot veure quants dels tuits amb una predicció errònia transmeten cada un dels sentiments restants.

5.1.2 Matriu de confusió en percentatges

Per visualitzar tant els percentatges d'encert com per analitzar com es distribueixen aquells tuits on s'ha fallat al fer la predicció de sentiment s'ha adaptat la matriu de confusió calculant els percentatges per columnes. Els valors de la matriu indiquen quin percentatge sobre tots els tuits predits amb un sentiment en concret correspon a cada sentiment realment.

		Prediccions	
		Positiu	**Negatiu**
Reals	**Positiu**	69,32%	24,24%
	Negatiu	30,68%	75,76%

Figura 10: Matriu de confusió en percentatges per 2 Clusters

La versió de l'algoritme que nomes classifica tuits positius i negatius a obtingut uns resultats satisfactoris al tenir un percentatge d'encert del 72,73%. Al comprovar l'encert individual de les prediccions dels tuits positius i negatius s'observen uns millors resultats a l'identificar tuits negatius, amb un 6,5% més d'encert. Aquesta diferencia, tot i que no implica una certesa, proposa certa facilitat de l'algoritme a identificar sentiments negatius respecte els positius.

	Prediccions			
Reals		**Positiu**	**Negatiu**	**Neutre**
	Positiu	44,53%	15,15%	32,08%
	Negatiu	22,63%	76,77%	37,74%
	Neutre	32,85%	8,08%	30,19%

Figura 11: Matriu de confusió en percentatges per 3 Clusters

La versió que també té en compte la possibilitat de trobar tuits neutres ha obtingut un percentatge total d'encert del 52,9%. Identificar correctament en mes

de la meitat dels casos a l'hora de classificar en 3 grups diferents, tot i no ser resultats ideals, demostra que el programa té un criteri. Es pot observar una gran diferència d'efectivitat en l'encert individual predient cada sentiment. El percentatge d'encert predient tuits negatius és del 76,77%, en contra, el percentatge d'encert predient tuits positius és només del 44,53% i el de neutres del 30,19%, un resultat inferior al percentatge d'aleatorietat[5].

En classificar-se en 3 grups, la matriu de 3 clusters també ofereix la informació de com es distribueixen els tuits mal predits. D'aquells tuits predits erròniament com a positius hi ha una quantitat significativament superior de neutres. Pel que fa als negatius erròniament predits, s'han predit quasi el doble com a positius dels predits com a neutres.

L'algoritme de 3 clusters, tot i que preten diferenciar els tuits neutres, obté un percentatge d'encert del 30,2% predient quins tuits seran neutres, un resultat inferior al percentatge d'aleatorietat. La distribució dels tuits dels quals s'ha fallat la predicció és prou homogènia entre els positius i negatius.

5.1.3 Relació de l'encert amb valor absolut del sentiment

Els tuits es classifiquen a partir d'un valor numèric que indica el sentiment de cada un, amb aquest valor de sentiment es pot interpretar com és d'intens el sentiment que transmet el tuit. A partir d'aquest valor es pot comprovar si la suposició de què com més intens és el sentiment que transmet un tuit, més alta serà la probabilitat que l'algoritme sigui capaç d'identificar correctament el sentiment que transmet.

Per comprovar la suposició s'utilitza una gràfica de barres on cada columna és el percentatge d'encert d'un conjunt de tuits amb una intensitat de sentiment. Aquests conjunts de tuits s'obtenen dividint la mostra de tuits etiquetats en 4 grups de la mateixa mida. Cada grup contindrà els tuits amb un rang concret del valor absolut del valor de sentiment, sent el rang del 0% al 25% aquell amb els tuits considerats com a menys intensos i el rang del 75% al 100% aquells que ho són més.

[5]El percentatge d'aleatorietat és el percentatge d'encert que s'obtindria si es fes la classificació aleatòriament, en classificar en 2 grups el percentatge d'aleatorietat és del 50%, per 3 grups és del 33,3%.

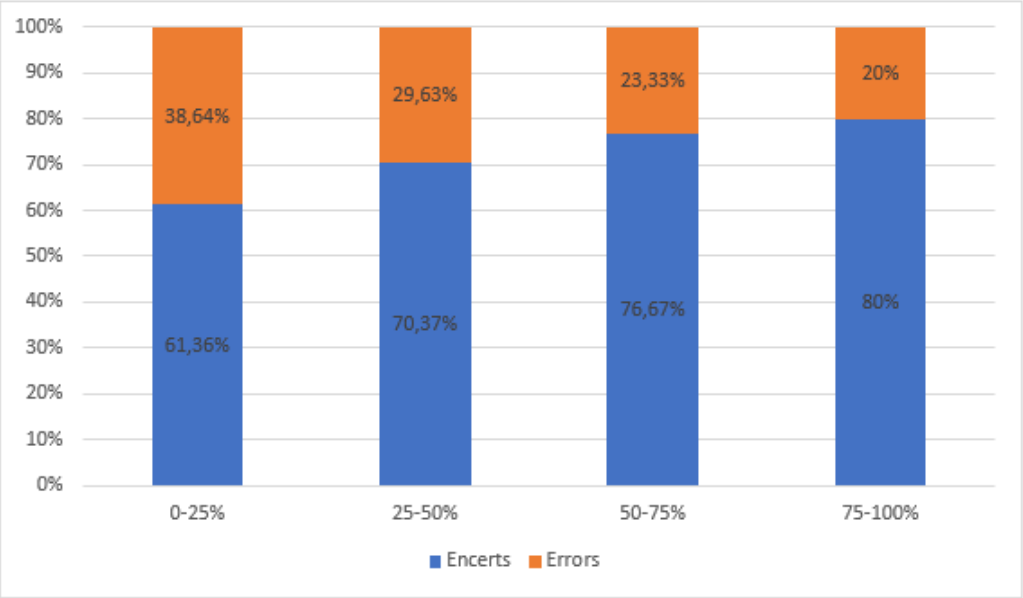

Figura 12: Encert per fases de 2 clusters

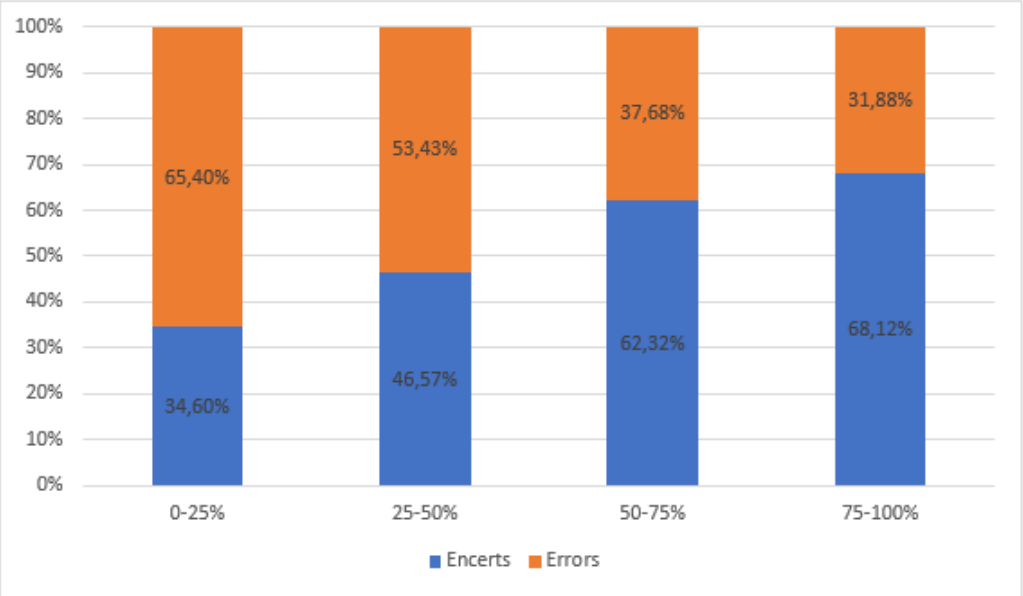

Figura 13: Encert per fases de 3 clusters

Les gràfiques d'ambdues versions mostren com a mesura que s'identifiquen els tuits més polaritzats, la probabilitat de què l'algoritme predigui el sentiment correctament creix amb constància, la suposició és correcte. També s'observa com en els rangs de polarització més baixa el creixement de la probabilitat d'encert és superior a la que s'observa en els rangs amb els tuits que transmeten més intensitat.

5.2 Resultats del Dataset

5.2.1 Ràtio de tuits positius i negatius, 2 clusters

Predicció	Mostra de 220 tuits		Total del Dataset	
	Número	%	Número	%
Positives	88	40%	134.442	43,33%
Negatives	132	60%	175.805	56,66%

Figura 14: Taula on es comparen els resultats entre la mostra i el tot el Dataset

Reals	Número	%
Positives	93	42,27%
Negatives	127	57,73%

Figura 15: Ràtio dels tuits etiquetats manualment

Per comprovar si la mostra de 220 és vàlida com a representació de tot el Dataset es comparen les ràtios de tuits positius i negatius de la mostra i de la totalitat del Dataset. Sent el marge d'error de 3,33%, la mostra es considera una representació lleial a la totalitat del Dataset. La ràtio de tuits positius i negatius que prediu l'algoritme per la mostra de 220 tuits només té un error del 2,27% respecte a la realitat.

5.2.2 Ràtio de tuits positius, negatius i neutres, 3 clusters

Predicció	Mostra de 289 tuits		Total del dataset	
	Numero	%	Numero	%
Positives	137	47,40%	152.316	49,09%
Negatives	99	34,26%	93.253	30,06%
Neutres	53	18,34%	64.678	20,85%

Figura 16: Taula on es comparen els resultats entre la mostra i el tot el Dataset

Realitat	Numero	%
Positives	93	31,18%
Negatives	127	43,25%
Neutres	69	23,87%

Figura 17: Ràtio dels tuits etiquetats manualment

La diferència entre els percentatges de predicció de la mostra de 289 tuits i la totalitat del Dataset no supera el 4,2% en cap de les 3 classificacions de sentiments, es considera una representació prou fiable. La comparació de les prediccions amb la realitat indica que l'algoritme de 3 clusters ha fallat en predir una ràtio de tuits positius, negatius i neutres semblant a la realitat. Mentre el percentatge de tuits neutres predits s'apropa a la realitat, els grups de positius i negatius tenen de mitja un error del 15,5%, un percentatge no menyspreable.

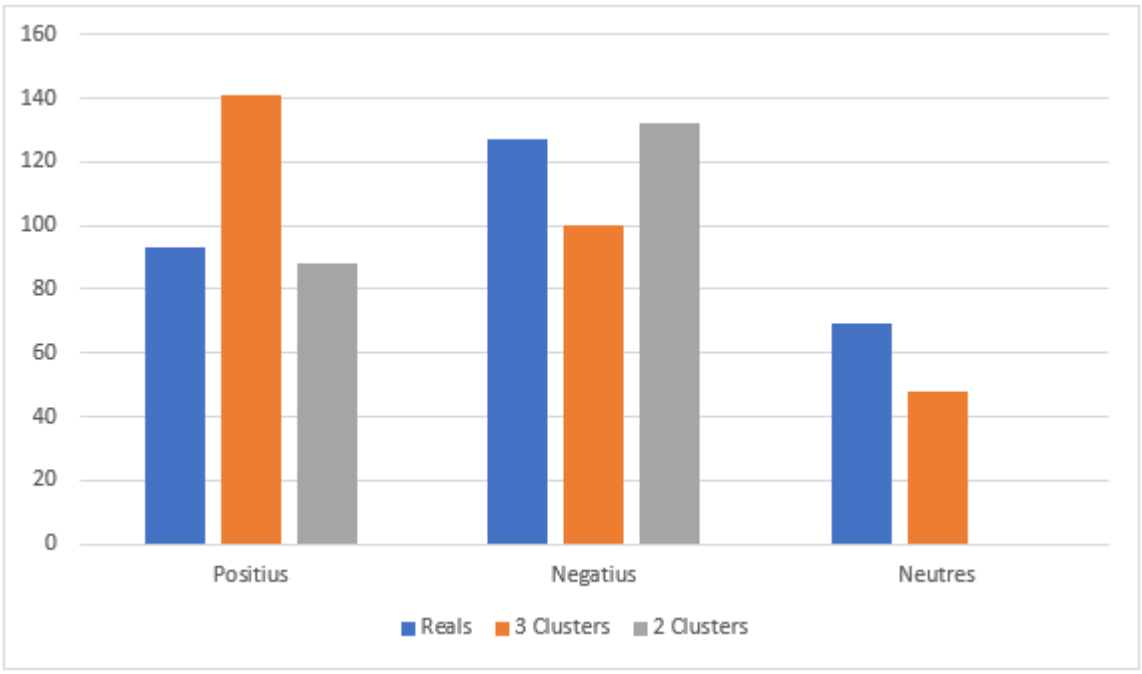

Figura 18: Gràfica que mostra la quants tuits de cada sentiment hi ha i el nombre de predits per 2 i 3 clusters.

Amb la gràfica es pot visualitzar com la quantitat de tuits de cada sentiment a la realitat i la predicció per l'algoritme de 2 clusters està anivellada. D'altra banda, la versió per 3 clusters té una quantitat de tuits significativament superior predita com a positiu i una quantitat inferior de tuits negatius i neutres.

Capítol 6

6 Conclusions

La versió de l'algoritme on els tuits només es classifiquen entre positius i negatius, 2 clusters, ha obtingut un percentatge d'encert del 72,7%. Tot i que l'encert de l'algoritme és satisfactori, no és ideal, l'algoritme és capaç de calcular una ràtio de tuits positius i negatius del Dataset aproximant-se a la realitat però amb un cert error. Un inconvenient d'aquest algoritme és que s'assumeix que no hi haurà tuits neutres al Dataset a analitzar. Al tractar-se d'un algoritme de baix cost i amb el que no cal etiquetar les dades, aquest pot ser útil per aquelles empreses que volen analitzar ràpidament i a grans trets l'acceptació d'un conjunt de tuits. En el cas de buscar una alta precisió en els resultats, aquest algoritme no serà suficient i caldrà utilitzar un alternatiu de tecnologia deep learning on serà necessari dedicar una gran quantitat de temps i recursos a etiquetar manualment un gran nombre de dades.

L'algoritme de classificació entre tuits positius, negatius i neutres ha obtingut un 52,9% d'encert. Dels resultats d'aquesta versió de l'algoritme contrasta la gran diferència d'encert que obté cada sentiment individualment amb un 76,8% d'encert els tuits predits com a negatius, un 44,5% els positius i un 30,2% els neutres. Obtenint un alt percentatge d'encert de tuits negatius destaquen els mals resultats al classificar els altres dos sentiments, especialment el neutre, amb un percentatge inferior al percentatge d'aleatorietat.

La suposició que motiva la implementació de la versió que identifica tuits neutres era que, encara que els percentatges d'encert totals segurament serien menors a la versió de 2 clusters pel fet d'haver de classificar entre 3 grups, les classificacions de tuits positius i negatius seria més encertada a l'haver apartat aquelles paraules neu-

tres. Veient els percentatges individuals d'encert de cada sentiment es conclou que la suposició no era correcta. A més, un percentatge d'encert inferior a l'aleatorietat en la predicció de tuits neutres i un encert inferior al 50% dels tuits positius descarta l'algoritme com a algoritme satisfactòriament funcional. Aquest fet es pot veure en la comparació de la ràtio de cada sentiment que hi ha a la realitat i la predicció que fa l'algoritme, on hi ha més d'un 15% d'error.

Els mals resultats a l'hora de detectar tuits neutres són deguts a la dificultat que suposa detectar paraules neutres, sent aquestes molt menys contundents i reconeixibles del que ho són aquelles paraules que denoten sentiment. Per aquest mateix motiu, els tuits negatius es detecten amb més encert en les dues versions de l'algoritme, les paraules negatives que s'utilitzen en els tuits negatius són les més contundents i, per tant, són les més fàcils de reconèixer per l'algoritme.

Un altre motiu dels mals resultats a l'identificar tuits neutres recau en la manera de concloure quins tuits són neutres. Amb les gràfiques on es relaciona el percentatge d'encert amb el valor absolut de sentiment s'aprecia que, com més petit sigui el valor absolut de sentiment d'un tuit, més complicacions tindrà l'algoritme per identificar correctament el sentiment que transmet. Per altra banda, l'algoritme detecta com a neutres aquells tuits que no són ni positiu ni negatius, aquells amb un valor absolut de sentiment inferior a un varem. Tots aquells tuits considerats neutres tindran un valor absolut de sentiment molt petit i, per tant, tots formaran part del rang on l'algoritme té més dificultats per identificar encertadament el sentiment.

6.1 Treball Futur

L'algoritme implementat en el projecte pot rebre diferents modificacions que millorin el rendiment.

6.1.1 Millorar la detecció de tuits neutres

L'algoritme de 3 clusters considera neutres tots aquells tuits que no destaquen ni com a positius ni com a negatius, implementar un mètode per detectar activament quins tuits són neutres milloraria els percentatges d'encert. Un possible plantejament pot ser tendir a zero el valor sentiment del tuit proporcionalment a la suma de valors de rellevància de les paraules neutres del tuit. Així, com més rellevància tinguin les paraules neutres en un tuit, més es reduiria el valor sentiment i, per tant,

les probabilitats d'entrar al rang de tuits considerats neutres augmentaria.

6.1.2 Modificar la mida mínim dels tuits

Analitzar com afecta modificar el filtratge pel nombre mínim de paraules dels tuits. Com més estricte sigui el filtre, més grans seran els tuits a entrenar, però més se'n descartaran, reduint el nombre de tuits del Dataset. El nombre mínim de paraules dels tuits és rellevant perquè al buscar similituds entre paraules per classificarles s'analitzen les paraules al voltant de cada paraula i, per tant, l'estructura del tuit serà un factor de gran importància. Com més grans siguin els tuits a analitzar més clares seran les estructures i amb més facilitat es trobaran les relacions entre paraules, en contra, menys tuits hi haurà per fer l'anàlisi, s'ha de buscar l'equilibri ideal.

6.1.3 Evitar tuits incomplets

Pel funcionament de la llibreria Tweepy, on el màxim de caràcters que retorna d'un tuit és 140, un percentatge dels tuits del Dataset no estaven complets, això ha pogut afectar en el rendiment del programa no podent analitzar la totalitat del contingut dels tuits. Utilitzar una eina amb la qual l'obtenció de la totalitat dels tuits sigui possible seria una millora del projecte.

6.1.4 Agrupar paraules per similitud de cosinus

Agrupar els vectors a partir de la similitud de cosinus ofereix una millor mesura de similitud que la distancia euclidiana. En aquest projecte s'utilitza la versió de K-means clustering que agrupa els vectors paraula a partir de la distància euclidiana al ser l'única versió de l'algoritme que ofereix scikit learn. Una millora a l'algoritme és fer l'agrupació amb la distància de cosinus, per fer-ho es pot buscar una implementació de l'algoritme d'agrupació, però també es pot convertir la distància euclidiana a una mesura proporcional de la distància de cosinus.

Bibliografia

[1] Nourah Alswaidan · Mohamed El Bachir Menai: A survey of state-of-the-art for emotion recognition in text approaches.
https://www.researchgate.net/publication/340010622_A_survey_of_stateoftheart_approaches_for_emotion_recognition_in_text

[2] Rafał Wójcik: Unsupervised Sentiment Analysis,
https://towardsdatascience.com/unsupervised-sentiment-analysis-a38bf1906483

[3] Gonzalo Ruiz de Villa: Introducción a Word2Vec,
https://gruizdevilla.medium.com/introducci%C3%B3n-a-word2vec-skip-gram-model-4800f72c871f

[4] Chris McCormick: Word2Vec Tutorial - The Skip-Gram Model,
https://medium.com/nearist-ai/word2vec-tutorial-the-skip-gram-model-c7926e1fdc09

[5] Edward Ma: How Negative Sampling work on word2vec?,
https://medium.com/@makcedward/how-negative-sampling-work-on-word2vec-7bf8d545b116

[6] Scikit-learn: K-means,
https://scikit-learn.org/stable/modules/clustering.html#k-means

[7] Scikit-learn: K-means documentation,
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[8] Bruno Stecanella: What is TF-IDF?,
https://monkeylearn.com/blog/what-is-tf-idf/

[9] Scikit-learn: TfidfVectorizer,
https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[10] Tweepy: Tweepy Document,
https://docs.tweepy.org/en/stable/l

[11] Gensim: Word2vec embeddings,
https://radimrehurek.com/gensim/models/word2vec.html

[12] Tanveer Khan: Relationship between Cosine Similarity and Euclidean Distance,
https://medium.com/ai-for-real/relationship-between-cosine-similarity-and-euclidean-distance-7e283a277dff