Mineria d'opinions a Twitter en un entorn multilingüe

Laia Mayol

Treball de fi de carrera
Consultor: Ramon Caihuelas

Enginyeria Tècnica en Informàtica de Gestió
Universitat Oberta de Catalunya
Barcelona, juny del 2013

Llistat de continguts

Llistat de figures

Llistat de taules

1 Introducció

La web és una gran mina d'opinions i sentiments: els usuaris poden valorar pel·lícules, restaurants, hotels, polítics, etc. Si fa 10 anys ens volíem comprar un cotxe, preguntàvem opinions al nostre entorn o, com a molt, consultàvem una revista especialitzada. Avui sembla impensable comprar-se un cotxe sense abans cercar comentaris d'altres usuaris sobre els models que ens interessen. Si una empresa vol saber si un determinat producte agrada als consumidors, hi ha una alternativa a les costoses enquestes: pot entrar en una xarxa social i veure què en diuen els usuaris del producte en qüestió. Tanmateix, extreure opinions de la web i tractar-les computacionalment és una tasca costosa i difícil. La mineria d'opinions (també anomenada anàlisi de sentiments) és el camp que s'ocupa de l'estudi computacional d'opinions, sentiments i emocions que s'expressen en un text.

L'objectiu d'aquest treball serà fer mineria d'opinions de la xarxa social de microblogging Twitter. En el 2012 Twitter tenia 500 milions d'usuaris registrats i es calcula que es generen uns 340 milions de tweets diaris. Tot això suposa una ingent quantitat dades que pot aportar informació molt valuosa. En concret, pel que fa a la mineria d'opinions és fàcil veure que té aplicacions immediates, especialment (tot i que exclusivament) en el món empresarial. Per exemple, algunes de les aplicacions podrien ser:

- Que una empresa pugui fer un estudi de mercat sobre un nou producte, de manera que obtingui dades fiables de les opinions que els consumidors estan expressant a Twitter.
- Que un partit polític pugui copsar la opinió pública sobre un determinat tema (una llei, un cas de corrupció, etc.)
- Hi ha empreses (o institucions o esdeveniments) que en la seva web corporativa han incrustat les mencions que altres usuaris fan de l'empresa en qüestió. L'objectiu de les empreses en fer això és fer-se publicitat mostrant que són un tema de conversa en una xarxa social. Tanmateix, què passa si apareix un tweet altament crític amb l'empresa? No seria millor aplicar un filtre per no mostrar els tweets molt negatius?[1]

[1] Gràcies al Ramon Caihuelas per donar-me la idea d'aquest objectiu.

Aquest treball s'estructura de la següent manera:

- El capítol 2 presenta un breu estat de l'art del tema de la mineria d'opinions.
- El capítol 3 presenta part de les dades amb què hem treballat.
- El capítol 4 presenta un model de mineria d'opinions basat en un lexicó simple.
- El capítol 5 presenta un model de mineria d'opinions basat en regles d'associació.
- El capítol 6 presenta un model de mineria d'opinions basat en *clustering*.
- El capítol 7 presenta els conclusions d'aquest treball.
- L'annex A presenta algunes de les regles extretes en el capítol 5.
- L'annex B presenta els scripts que s'han fet servir per extreure i tractar les dades i fer les tasques de mineria.

2
Breu estat de l'art

En els últims anys s'ha produït una explosió de treball sobre anàlisi d'opinions i sentiments. Segons Liu (2010), una opinió o sentiment es pot definir formalment com la següent tupla:

(1) *o, tr, oo, e, t*
On o és l'objecte, tr un tret (o característica) de l'objecte, oo és l'orientació de l'opinió sobre el tret t de l'objecte o (pot ser positiva o negativa), e és l'emissor de l'opinió i t el temps en què s'ha omès l'opinió.

Un objectiu de mineria d'opinions podria ser trobar totes les tuples que apareguin en un determinat text. Per exemple, considerem el text següent:

(2) (1) Ahir em vaig comprar un iPhone. (2) És una passada. (3) La pantalla tàctil és genial i la qualitat de veu també és molt bona. (4) L'única pega és que la bateria no dura gaire. (5) La meva mare, però s'ha enfadat amb mi per haver-lo comprat i diu que és massa car.

En aquest text trobem diverses opinions sobre diversos trets i de diversos emissors. Mentre que en la frase 2 l'escriptor del text expressa una opinió general positiva sobre l'objecte (el telèfon iPhone), les frases 3 i 4 expressen opinions sobre diversos trets del telèfon (la pantalla, la qualitat de veu i la bateria). Finalment, la frase 5 expressa una opinió negativa del preu de l'objecte. Tanmateix, aquesta opinió no és emesa per l'escriptor del text, sinó per la seva mare.

La tasca anteriorment descrita és molt complexa, ja que suposa fer una anàlisi lingüística molt detallada. Per això, normalment les tasques de mineria d'opinions simplifiquen algun dels paràmetres de la tupla. Les principals tasques que es duen a terme dins del camp de la mineria de dades són les següents.

1. Classificació de sentiments i subjectivitat: Aquesta tasca consisteix en classificar un text o bé segons si transmet una opinió positiva o negativa, o bé també segons si és subjectiu o objectiu. La classificació de sentiments serà la tasca que durem a terme en aquest treball.

2. Anàlisi de sentiments basada en trets: Aquesta tasca consisteix en classificar els diversos components o trets d'un objecte sobre el qual s'emet una opinió. Per exemple, tal i com vèiem en l'exemple (2), el parlant emet una opinió positiva de la pantalla i una opinió negativa de la bateria. Per tant, en aquesta tasca s'hauran d'extreure tant els trets com les opinions.

3. Anàlisi de sentiment de les oracions comparatives: De vegades no s'emet directament una opinió positiva o negativa, sinó que s'emet en forma de comparació, com en l'oració "La qualitat de la càmera x és millor que la de la càmera y". Aquesta tasca consisteix en fer mineria d'aquesta mena d'opinions comparatives.

4. Cerca d'opinions: Aquesta tasca consisteix en fer una cerca i recuperació de documents opinatius (negatius o positius) sobre un tema. D'una banda, caldrà cercar documents que són rellevants per la consulta (tal i com fa un cercador convencional) i de l'altra determinar si aquests documents expressen opinions i de quin tipus són.

5. Detecció d'spam d'opinions: Aquesta tasca consisteix en detectar spam d'opinió, és a dir comentaris falsos ja sigui per afavorir o perjudicar un producte, empresa o persona determinats.

2.1 Classificació d'opinions

Com hem comentat abans, en aquest treball ens centrarem en la primera tasca citada: la classificació de sentiments. Més formalment la tasca es pot resumir així:

(3) Donat un document que parla d'un objecte o, cal determinar l'orientació oo de la opinió expressada sobre o.

Per dur a terme aquesta tasca, s'han emprat sobretot mètodes d'aprenentatge supervisat amb l'objectiu de classificar els documents en dues categories (positius i negatius). Les dades per fer el training i el testing venen de pàgines de comentaris de productes, on el mateix escriptor del text classifica la seva opinió sobre el producte en una escala. Normalment, una recomanació amb 4 o 5 estrelles es considera positiva, i una recomanació amb 1 o 2 estrelles es considera negativa.

Els mètodes més emprats han estat amb classificadors bayesians (*naïve Baysean*) i amb màquines de vector de suport (SVM). Només considerant els unigrames (les paraules aïllades)

que apareixen en els comentaris aquestes tècniques ja aconsegueixen resultats força decents, com en treball de Pang et al. (2002). Treballs posteriors han inclòs més trets en l'anàlisi, com per exemple:

- Els termes i la seva freqüència. És a dir, les paraules més freqüents potencialment ens seran més d'ajuda a classificar un text. Com que nosaltres treballarem amb texts molt curts, és poc probable que aquest tret ens ajudi gaire.
- La categoria morfològica: els adjectius són indicadors importants de l'opinió i, per tant, sovint reben un tractament especial.
- Paraules i sintagmes d'opinió. Les paraules d'opinió són paraules que expressen opinions o sentiments positius o negatius. Per exemple, *fantàstic* i *espectacular* comuniquen opinions positives i *espantós* i *dolent* comuniquen opinions negatives. La majoria de paraules d'opinions són adjectius i adverbis, però certs noms (*merda*, *passada*), verbs (*encantar*, *agradar*) o expressions (*costar un ronyó*) també poden transmetre opinions.
- Dependències sintàctiques: alguns treballs han intentat emprar les relacions sintàctiques entre paraules per extreure informació.
- Negació. Òbviament la negació és important perquè la seva presència pot canviar l'orientació d'una opinió.

Apart de la classificació en opinions positives o negatives, també s'han dut a terme tasques de predicció de les valoracions d'un producte (per exemple, en una escala de l'1 al 5), com en la feina de Pang i Lee (2005). En aquest cas, la tasca es pot plantejar com un problema de regressió. Cal també tenir en compte que l'anàlisi d'opinions és molt sensible al domini: no és el mateix parlar de pel·lícules que de cotxes, per exemple. Per exemple, tal i com va observar Turney (2002), l'adjectiu *imprevisible* pot ser negatiu en un comentari sobre un cotxe ("el seu comportament és imprevisible) i positiu en un comentari sobre una pel·lícula ("l'argument és del tot emocionant i imprevisible").

Els mètodes no supervisats no s'han emprat tant, però tot i així el treball de Turney (2002) proposa la següent tècnica:

1. Extreure sintagmes que continguin adjectius o adverbis, ja que són els elements que solen comunicar les opinions.
2. Estimar l'orientació de l'opinió (*oo*) de cada sintagma fent servir la següent fórmula:

(4)
$$log_2 \left(\frac{hits(sintagmaNEAR"excellent")hits("poor")}{hits(sintagmaNEAR"poor")hits("excellent")} \right)$$

La idea bàsica és que s'estima si el sintagma en qüestió està més associat amb una paraula positiva o amb una paraula negativa. Aquesta associació es computa mirant les vegades, que en un cercador, la paraula en qüestió apareix a prop de la paraula *excellent* o a prop de la paraula *poor*.

3. Calcular la mitjana de l'orientació d'opinió de tots els sintagmes extrets d'un text. Si és positiva es classifica el text com a positiu, si no ho és es classifica com a negatiu.

2.2 Generació d'un lexicó d'opinió

Per poder fer una bona de classificació, és sovint bàsic disposar d'un bon lexicó d'opinió. És a dir, disposar d'una llista de paraules que solen remetre a opinions positives i una altra llista de paraules que comuniquen opinions negatives. Un cop hem compilat manualment aquests llistes, hi ha diverses maners d'ampliar-les.

- Mètodes bastats en diccionaris: Aquests mètodes consisteixen en fer *bootsraping* bastant-nos en un conjunt de paraules anotades manualment i un diccionari. La principal estratègia és fer servir la informació dels diccionaris sobre sinònims i antònims per ampliar la llista generada manualment. Els sinònims i antònims s'afegeixen a la llista i es fa una altra iteració per cercar nous sinònims o antònims. El procés acaba quan ja no trobem paraules noves. El principal inconvenient d'aquest mètode és que no trobarà paraules d'opinió que siguin específiques d'un domini. En canvi, els mètodes basats en corpus no tenen aquest problema.

- Mètodes basat en corpus: Aquests mètodes consisteixen en agafar un conjunt de paraules anotades manualment i fer-la servir per identificar més paraules d'opinió en un corpus fent servir certes construccions lingüístiques. Per exemple, una d'aquestes construccions és la coordinació. Normalment els adjectius que es coordinen amb la conjunció *i* comparteixen orientació (o bé tots dos són positius o bé tots dos negatius) i, en canvi, els que es coordinen amb la conjunció *però* tenen orientacions diferents.

2.3 Objectius del TFC

Com hem vist, la majoria dels treballs previs en el camp de la mineria d'opinions tenen dues propietats: fan un treball d'aprenentatge supervisat i treballen a partir de l'anglès. De fet, té sentit que sigui així ja que per l'anglès existeixen una gran quantitat de dades anotades en tota mena de plataformes com IBDM per pel·lícules, Tripadvisor per hotels i restaurants i Amazon per tota mena de productes. En aquest treball, ens apartarem d'aquesta línia de treball en dos aspectes. Farem servir dades no supervisades de Twitter i treballarem amb dades produïdes a Catalunya i, per tant, majoritàriament bilingües en català i castellà.

El català es troba poc present (o directament absent) de xarxes socials amb dades supervisades com IMDB, Amazon o Tripadvisor. En canvi, la seva presència és notable a Twitter. D'altra banda, Twitter suposa una enorme font d'informació on la gent comenta, opina i s'expressa en temps real, cosa que el converteix en una plataforma òptima per fer mineria de dades i més concretament mineria d'opinions. Per tant, el nostre objectiu de mineria de dades serà extreure opinions de diferents temes a partir de tweets escrits a Catalunya.

3
Extracció de dades

Per facilitar l'extracció de dades de la xarxa social Twitter s'ha fet servir el software R (RDevelopment, 2011) i el paquet twitteR (Gentry, 2013) que proporciona una interfície amb l'API de Twitter. Concretament té una funció de cerca (*serachTwitter*) que permet cercar tweets que compleixin unes determinades condicions: que continguin una determinada seqüència, que hagin estat escrits en un determinat lloc, en una determinada llengua, etc. Com que el català no és una de les llengües que apareguin en la metadata de Twitter, hem optat per cercar tweets escrits en un radi de 70 km de Manresa (amb la qual cosa cobrim bona part del territori de Catalunya). Hem cercat paraules clau de tres temes: (i) temes d'interès social (específicament relacionades amb la crisi dels desnonaments que s'ha viscut durant els últims mesos), (ii) marques comercials (de tecnologia o companyies aèries) o (iii) cadenes de supermercat.

A l'annex B.1, es pot veure l'script per baixar i començar a tractar les dades. Les dades extretes han estat les següents:

- 1500 tweets[1] que contenen la paraula 'Apple'.
- 1500 tweets que contenen la paraula 'Samsung'.
- 1500 tweets que contenen la paraula 'Iberia'.
- 1500 tweets que contenen la paraula 'PAH' (Plataforma d'Afectats per la Hipoteca)
- 1500 tweets que contenen la paraula 'Colau' (cognom de la portaveu de la PAH).
- 1500 tweets que contenen la paraula 'Escrache'.
- 1500 tweets que contenen la paraula 'Caprabo'.
- 1500 tweets que contenen la paraula 'Mercadona'.
- 1500 tweets que contenen la paraula 'Carrefour'.

[1] El límit de 1500 ve imposat pel motor de cerca de Twitter sobre quants tweets es poden recuperar a la vegada.

Un cop obtinguts els tweets, hem extret el text en si del tweet (descartant la metadata) i hem fet una mica de tractament simple per tal de poder-lo processar. En concret, en baixar-nos les dades apareixen alguns caràcters que no formen part del UTF-8 i que ens impedeixen tractar les dades correctament. Per evitar aquesta situació hem eliminat tots els caràcters que no formin part del UTF-8 i hem substituït totes les vocals accentuades per vocals no accentuades.

Cal també mencionar que cadascun dels conjunts de dades poden tenir tweets repetits. Això es dóna si un tweet és enviat també per un altre usuari (és a dir, si és retweetejat). Considero que això no és un problema, sinó al contrari. Si, per exemple, moltes persones redifonen un tweet que emet una opinió negativa sobre una determinada marca, en la majoria de les ocasions voldrà dir que comparteixen aquesta opinió.

4
Mineria d'opinions amb un lexicó simple

4.1 Mètode

En aquest capítol emprarem la tècnica descrita a Breen (2011) per fer mineria d'opinions a partir de tweets. Bàsicament necessitem un lexicó de paraules positives i negatives. Com a punt de partida, hem agafat el lexicó disponible per l'anglès compilat per Hu i Liu (2004) i disponible en línia. Aquest lexicó conté unes 6800 paraules organitzades en dos fitxers: paraules negatives i paraules positives. En un primer moment, hem traduït automàticament aquests fitxers en català i castellà mitjançant Google Translate. Tanmateix, la qualitat de la traducció no era gaire bona, cosa que ens donava un lexicó de baixa qualitat. Per millorar la qualitat del lexicó, hem partit del lexicó del castellà SOL (Martínez-Cámara et al., 2013), que també ha estat generat a partir de traduir automàticament el lexicó de Hu i Liu i posteriorment revisar-lo automàticament. A més, també hem traduït aquest lexicó al català automàticament. Després, hem substituït les vocals accentuades per vocals no accentuades, tal i com hem fet en les dades. Finalment, hem acabat en un lexicó bilingüe català-castellà que conté 6296 paraules negatives i 2790 paraules positives[1].

En segon lloc, per a cada tweet hem mirat si contenia paraules d'opinió negatives o paraules d'opinió positives: cada paraula positiva suposa un valor de +1 i cada paraula negativa un valor de -1. D'aquesta manera, podem calcular la puntuació final de cada tweet. Si té una puntuació superior a 0, podem suposar que expressarà una opinió positiva (ja que conté més paraules positives que negatives). Si té una puntuació inferior a 0, expressarà una opinió negativa. Si té una puntuació igual a 0, o bé expressava una opinió neutra (contenia tantes paraules positives com negatives) o bé no expressa una opinió (o més ben dit, una opinió que puguem detectar amb un lexicó simple).

A l'annex B.2, es pot veure l'script que hem fet servir per fer mineria amb un lexicó simple.

[1]És interessant constatar que el lexicó conté més del doble de paraules negatives que positives, i podríem especular si això vol dir que les llengües tenen més paraules per expressar emocions negatives que positives. En tot cas, cal tenir en compte que això pot afectar els resultats.

4.2 Resultats

A la taula 4.1 podem veure les mitjanes de les puntuacions que trobem a les dades corresponents a les marques comercials i a la figura 4.1 podem observar la distribució en forma de histogrames d'aquestes puntuacions. Podem observar que a grans trets Apple és la marca més ben valorada, seguida per Samsung i, en últim lloc, per Iberia.

Marca	Puntuació
Apple	0.3
Iberia	-0.33
Samsung	0.05

Table 4.1: *Mitjanes de puntuació de Apple, Iberia i Samsung*

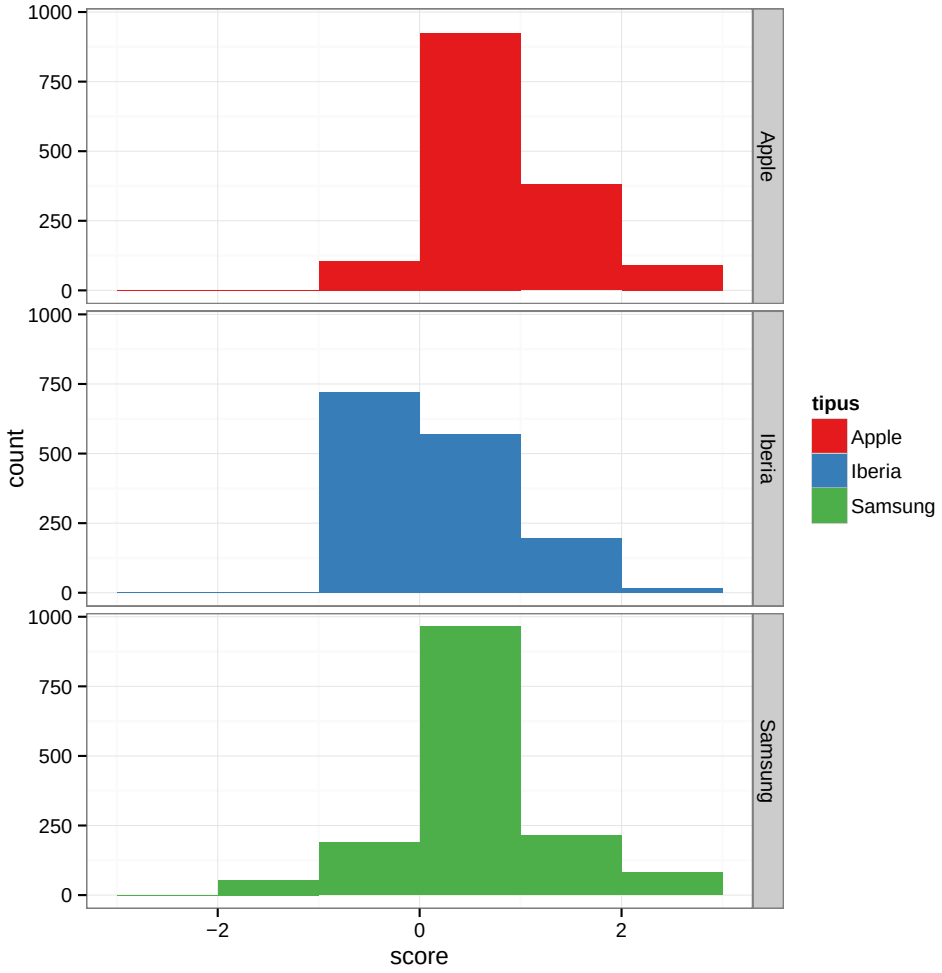

Figure 4.1: *Histogrames de les dades de Apple, Iberia i Samsung*

A la taula 4.2 podem veure les mitjanes de les puntuacions que trobem a les dades corresponents a la temàtica social i a la figura 4.2 podem observar la distribució en forma de histogrames d'aquestes puntuacions. Podem observar la mitjana per 'Colau' és positiva,

mentre que és negativa per 'PAH' i 'Escrache'.

Marca	Puntuació
Colau	0.24
Escrache	-0.13
PAH	-0.18

Table 4.2: *Mitjanes de puntuació de Colau, Escrache i PAH*

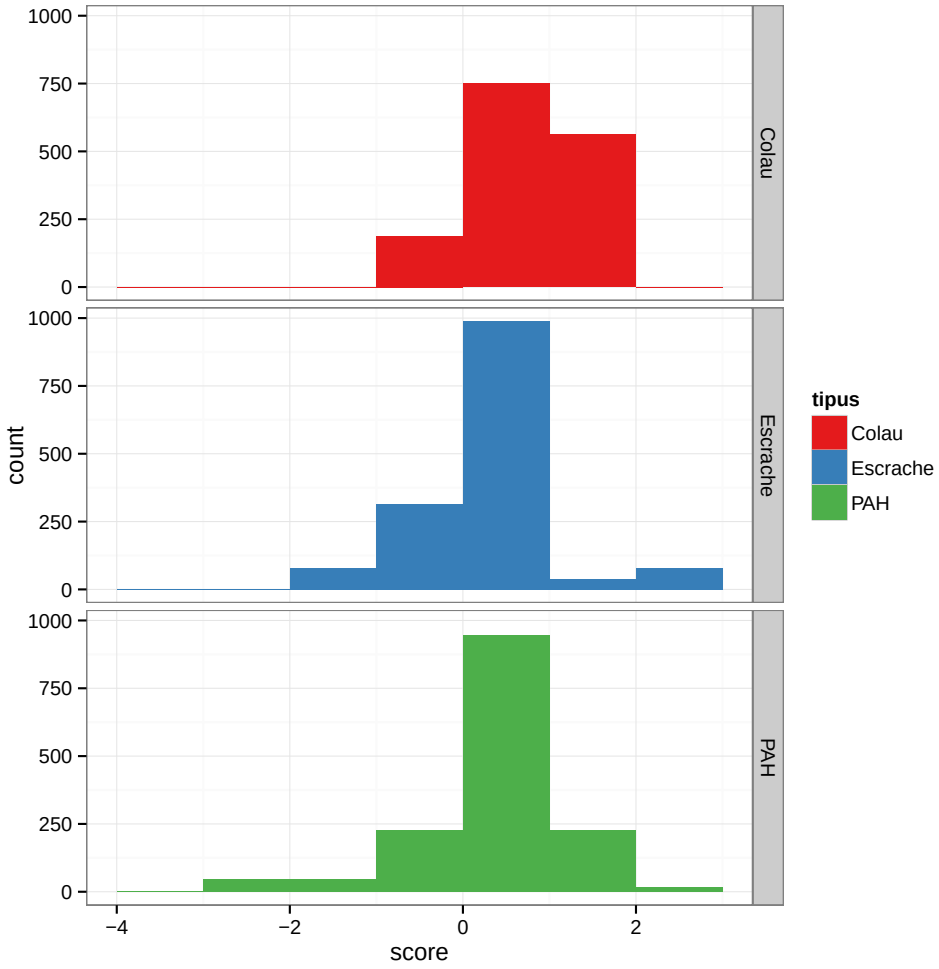

Figure 4.2: *Histogrames de les dades de Colau, Escrache i PAH*

A la taula 4.3 podem veure les mitjanes de les puntuacions que trobem a les dades corresponents als supermercats a la figura 4.3 podem observar la distribució en forma de histogrames d'aquestes puntuacions. Podem observar la mitjana és tocant a 0 per 'Caprabo' i lleugerament positiva per Carrefour i Mercadona.

Marca	Puntuació
Caprabo	-0.03
Carrefour	0.17
Mercadona	0.19

Table 4.3: *Mitjanes de puntuació de Caprabo, Carrefour i Mercadona*

4.3 Discussió

En aquesta secció analitzarem amb una mica més de detall les dades provinents dels tweets sobre marques comercials i comentarem breument els de temàtica social.

Com hem comentat abans, les dades mostren que a grans trets Apple és la marca més ben valorada, seguida per Samsung i, en últim lloc, per Iberia. És important destacar que fent servir aquest mètode tan senzill ja obtenim algunes impressions de quines marques són millor o pitjors valorades, tot i que també és evident que un mètode tan poc sofisticat només ens pot donar una impressió i no dades amb un grau elevat de fiabilitat.

Una de les primeres observacions que podem fer és que dels 13500 tweets analitzats, 7889 (un 58%) obtenen puntuació 0. És òbviament possible que molts d'aquests tweets no expressin opinió, però un anàlisi superficial de les dades ens mostra com alguns tweets que sí que expressen emoció reben puntuació 0, com podem veure a (1). Mentre que (1-a) expressa una opinió negativa sobre Samsung, (1-b) n'expressa una positiva. Tanmateix, com que no contenen cap paraula que figuri en el lexicó, acaben amb una valoració de 0.

(1) a. Samsung Galaxy 4 no es compatible con las TecTiles, anuncian TecTiles
b. RT @FerranGallofre: 1h i 40 min per actualitzar el software del IPhone.... Molt em temo que Samsung esta trucant a la porta...

Vegem ara una petita mostra de tweets segons si han estat ben classificats o mal classificats. Comencem pels ben classificats. A (2) podem observar una petita mostra de tweets ben classificats com a negatius i a (3) una mostra de tweets ben classificats com a positius.

(2) a. Virus simpáticos que te hacen formatear. Qué mono eh?? #Samsung #laptop #portátil #virus
b. RT @Uilifoc2014: Iberia fa 199 vols intercontinentals setmanalment des d Barajas. Només 2 desde Barcelona #MejorHundidos @ColonosCs.
c. La opinion del androide: Menos Samsung y mas otros

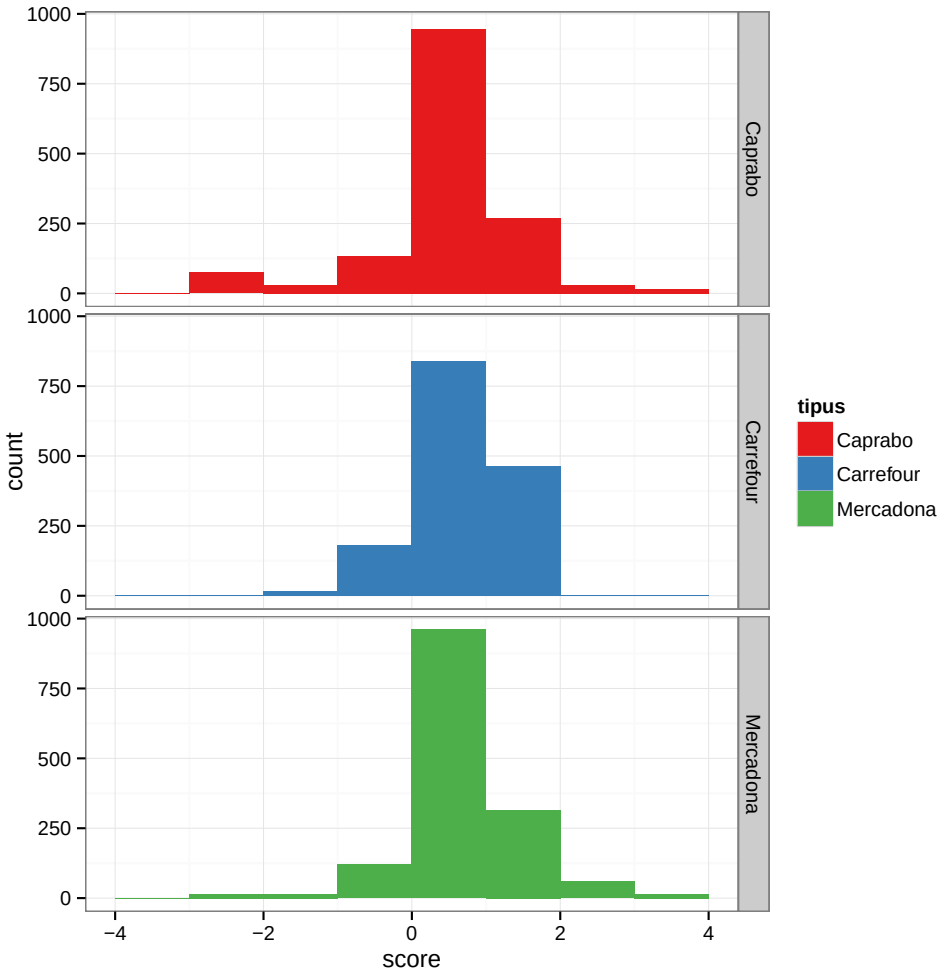

Figure 4.3: *Histogrames de les dades de Caprabo, Carrefour i Mercadona*

(3) a. Nova adquisició! A millorar la experiència de joc :) #logitech #samsung #philips #gamer

b. Caramba !!! @RumiGpp: Todo se redujo a un solo objeto #revolucionario #Apple #iPhone

c. RT @modastylo: "@MarketingInf: Apple ofrece 10.000 dólares de recompensa si alcanza los 50.000 millones de descargas

A l'exemple (2-a), la presència del mot *virus* ajuda a classificar el tweet correctament com a negatiu, a pesar de la presència del mot positiu *simpáticos* A (2-b), ens podríem plantejar si l'opinió negativa que s'expressa es dirigeix realment a Iberia. Cal també dir que aquest tweet és responsable en gran part de la baixa puntuació d'Ibèria, ja que va ser redifós moltes vegades.

Vegem ara una petita selecció de tweets mal classificats. Els tweets de (4) han estat classificats com a negatius tot i que propiament no expressen una opinió negativa. Tanmateix,

mirant els mots que hi apareixen és clar perquè han estat classificats així: hi apareixen paraules com *colgado*, *reparar*, *rompió* o *rivalitat*.

(4) a. @SamsungEspana Solucion problema Samsung SII que queda colgado: descargar ap "Wake Lock" y escojer "partialwakelock" #samsungs2
b. @alvalumi Busco cotizacion para reparar una laptop Samsung RV511, que se rompio la pantalla, saludos.
c. Anunci bastant original del nou telefon Nokia que s'enriu de la rivalitat Samsung-Apple.

A (5) tenim alguns tweets catalogats com a positius. El primer ha estat classificat com a positiu ja que la paraula *menor* apareix com a positiva en el lexicó, dins l'expressió *de menor precio*. Per tant, aquest error ve causat pel lexicó. El segon ha ha estat classificat com a positiu ja que *val la pena* també apareix com a positiva en el lexicó. Aquí, el problema és que el nostre mètode no té en compte que la negació canvia l'orientació de les paraules. Finalment, el tercer és un tweet irònic en què es comenta la publicitat que un tennista va fer de l'empresa Samsung des d'un iPhone. És molt difícil que un sistema automàtic pugui arribar a detectar la intenció irònica de l'autor d'aquest missatge i, per tant, és un error difícilment evitable.

(5) a. Apple lanza iOS 6.1.4 para iPhone 5, una actualizacion menor
b. Dega del colegi d'enginyers informatics de Catalunya: "Apple es una plataforma morta. No val la pena programar"
c. Si es que... @ElHuffPost: David Ferrer feliz de tener un Samsung... como cuenta desde su iPhone

Pel que fa a les dades sobre temàtica social, podem observar, com hem mencionat abans que expressen opinions més aviat negatives. A (6) podem veure una petita mostra de tweets que contenen la paraula *escrache*

(6) a. "Activistas de la PAH realizan un escrache en la casa de Moragas" sed malos [Veis como s. Desobediencia civil]
b. @surfzone he estat a punt de fer-li "escrache" pero, la seva cara de merdós quan m'ha mirat m'ha fet angúnia i m'he bloquejat.
c. @Pablo89 Poco criterio de la gente, escándalo es un escrache y no eso de irse con 88 millones, es que.....

El que podem observar és que l'opinió negativa que emeten aquests textos no és sobre el fenomen de l'escrache en si, sinó més aviat sobre els fets que propicien que es dugui a terme una activitat d'aquesta mena. És a dir, com que un escrache és una activitat de protesta contra alguna cosa s'associa amb esdeveniments negatius. Per tant, aquí la mineria no és serveix per detectar quina és la opinió pública sobre l'escrache, sinó més aviat per detectar que hi ha un descontentament social.

4.4 Conclusió

En aquest capítol, hem presentat un model de mineria d'opinions basat en un lexicó simple. Tot i que el model presenta alguns problemes (hi ha tweets positius catalogats com a negatius, tweets negatius catalogats com a positius, i un percentatge elevat de tweets que queden sense classificar), aquest model permet fer-me una primera idea de la opinió general sobre un determinat tema. A més és un mètode poc costós computacionalment i, per exemple, per filtrar certs missatges negatius i evitar que surtin en pàgines webs corporatives es podria aplicar fàcilment.

En el següent capítol mirarem de fer servir una tècnica més clàssica de mineria de dades: les regles d'associació.

5
Regles d'associació

En aquest capítol aplicarem el mètode de les regles d'associació per fer mineria en les dades extretes de Twitter.

Una regla d'associació és una expressió de la forma següent (Sangüesa i i Luis Molina, 2012):

(1) $X \Rightarrow Y$,
on X i Y són conjunts d'elements que poden prendre valors binaris.

Una regla d'associació expressa que si apareix X en un conjunt de dades també hi sol aparèixer Y.

Els dos principals paràmetres que cal tenir en compte en interpretar una regla d'associació és la confiança i el suport:

- Una regla d'associació té una confiança c si el c% dels ítems del conjunt de dades analitzat que contenen X també contenen Y.
- Una regla d'associació té un suport s si el s% dels ítems del conjunt de dades analitzat contenen $X \cup Y$.

El nostre objectiu serà veure amb quina mena de paraules s'associen cadascun dels temes dels quals tenim dades. És a dir, s'associa "Apple" amb unes determinades paraules? Si una marca s'associa amb les paraules *fantàstic* i *car* segurament podrem treure unes conclusions diferents de si s'associa amb *dolent* i *cutre*.

5.1 Preparació de les dades

Per poder dur a terme la mineria amb regles d'associació caldrà preparar les dades. En concret, caldrà dur a terme els passos següents:

1. Netejar el text dels tweets eliminant espais en blanc de més, signes de puntuació, números, etc. de manera que ens quedi un llistat de paraules separades per un sol espai en blanc.

2. Treure les paraules buides dels tweets. Les paraules buides són aquelles que no aporten significat lèxic, sinó gramatical. És a dir, no remeten a cap concepte. Per exemple, els articles i les preposicions són paraules buides. Les paraules buides són molt freqüents en un text, però no aporten informació semàntica. Per tant, les haurem d'eliminar dels tweets.

3. Binaritzar les dades, de manera que ens quedi una taula amb el següent format, on per cada paraula indiquem si aquella paraula apareix (marcat amb '1') o no apareix (marcat amb '0') a cada tweet:

Tweet	Paraula #1	Paraula #2	Paraula #3
Tweet 1	1	1	0
Tweet 2	0	0	1
Tweet 3	0	1	0

Table 5.1: *Exemple de dades binaritzades*

Aquestes tres tasques es poden dur a terme amb el package de R *tm* (Feinerer, 2012). A l'annex B.3, es pot veure l'script que hem fet servir per extreure les regles d'associació.

5.2 Resultats

Per dur a terme la tasca de mineria, hem fet servir l'algoritme 'Apriori' del package de R *arules* (Hahsler et al., 2007). He extret les regles amb un valors força baixos tant de confiança com de suport: un mínim de confiança de 0.5 i un mínim de suport de 0.2.

- Mercadona: Es troben 30 regles (vegeu apèndix). Totes venen d'un sol tweet, que podem veure a (2), que va ser molt retweetejat. Cap de les regles no ens dóna informació significativa sobre la marca.

  (2) España debería ir a Eurovision con la canción del Mercadona.

- Caprabo: Es generen 0 regles amb els mínims específicats.

- Carrefour: Es generen 150 regles (vegeu apèndix). Totes venen d'un sol tweet, que podem veure a (3), que va ser molt retweetejat. Cap de les regles no ens dóna informació

significativa sobre la marca.

(3) BEN AFFLECK ES MEJOR COMO DIRECTOR QUE COMO CAJERO DEL CARREFOUR.

- Apple: Es generen 0 regles amb els mínims específicats.
- Samsung: Es genera la següent regla:

(4) samsung $\Rightarrow$ =galaxy 0.4793333 0.5161522 1.076813

La regla no ens dóna informació directa sobre les opinions que tenen els usuaris d'aquesta marca, però sí que ens diu que el producte de Samsung del qual més es parla és el model Galaxy.

- Iberia: Es generen 4592 regles (vegeu apèndix). Tot i que l'elevat nombre, totes venen d'un mateix tweet, que podem veure a (5), que va ser molt retweetejat.

(5) RT @Uilifoc2014: Iberia fa 199 vols intercontinentals setmanalment des d Barajas. Només 2 desde Barcelona #MejorHundidos @Colonos$_{C}s$

- PAH: Es generen 0 regles amb els mínims específicats.
- Escrache: Es generen 882 regles (vegeu apèndix). Tot i que l'elevat nombre, totes venen d'un mateix tweet, que podem veure a (6), que va ser molt retweetejat. Cap de les regles ens dóna informació útil.

(6) RT @DRYGirona: RT @Psicodromo: Yo tambien hago #Escrache en este video: http://t.co/MUaAw4Rwhz es #PÁHrajoy http://t.co/SoMVGA2iVv

- Colau: Es generen 4696 regles (vegeu apèndix). Tot i que l'elevat nombre, totes venen d'un mateix Tweet, que podem veure a (7), que va ser molt retweetejat. Cap de les regles ens dóna informació útil.

(7) Ada Colau arremete de nuevo contra TVE: 'Menudo asco de Informe Semanal' http://t.co/lbhytSOnzX

5.3 Conclusions

Podem concloure que l'ús de regles d'associacions no ens ha permès extreure cap mena de coneixement de les nostres dades. El principal problema que ens hem trobat és que disposem de relativament poques dades i això fa que només que hi hagi un tweet que ha estat força redifós, aquest tweet determini les regles que es crearan (que bàsicament determinaran que hi ha una certa relació entre les paraules d'aquest tweet concret).

És plausible pensar que si disposessim de moltes més dades haguessin pogut sorgir algunes regles interessants que sí que ens donessin informació sobre la marca o el fenomen en examen.

En el següent capítol farem servir una altra de les tècniques clàssiques de la mineria de dades i, a més, també canviarem les dades amb què treballem, cosa que ens farà canviar una mica l'enfocament.

6
Clustering

En aquest apartat hem pres un enfocament una mica diferent del presentat en els capítols anteriors. No ens hem centrat en tweets que parlen d'una marca o fenòmens, sinó que hem extret dades que contenien paraules positives i negatives i hem aplicat tècniques de clustering per veure si podem extreure automàticament relacions entre paraules o entre documents que contenen certes paraules. La idea és veure si els models de clústering poden veure que un tweet positiu té més coses en comú amb un altre tweet positiu que amb un de negatiu i agrupar-los en el mateix clúster.

Els models que clustering tenen com a objectiu agrupar objectes amb característiques semblants sense cap imposar cap criteri *a priori* (vegeu, per exemple l'explicació a Sangüesa (2012)). El resultat final d'un model de clustering és la divisió dels objectes en particions (o clústers), tal que tot objecte es trobi en un un sol clúster. Les tècniques de clústering solen computar una mesura de la distància entre objectes, de manera que s'agrupin en un sol clúster els objectes propers i quedin situats en clústers diferents els objectes llunyans.

Un dels mètodes de clústering més emprats, i que empraré per fer clústers de tweets, és el mètode dels centroides, també anomenat k-means. Sense entrar en gaires detalls, aquesta tècnica funciona de la següent manera. Primer, es determinen el nombre de clústers k que voldrem obtenir, després es trien k objectes que seran les llavors de cada clúster i es calculen els centres de cada clúster. En tercer lloc, s'assignen els objectes al grup que tingui el centre més proper. Es re-calculen els centres i es segueix iterant fins que ja no hi ha canvis.

Un altre dels mètodes de clústering que provarem serà un mètode de clústering jeràrquic, anomenat de Ward. Els mètodes jeràrquics tenen com a objectiu crear una jerarquia de clústers, de manera que puguem veure quines relacions hi ha entre ells. El mètode de Ward és un mètode jeràrquic aglomeratiu en què cada observació comença el seu propi clúster. Després anem agregant parelles de clústers, fins que totes les observacions n'han creat un de sol.

Primer començaré comentant el procés d'extracció i preparació de dades. Seguidament presentaré un model de clusterització de paraules i, finalment, un model de clusterització de tweets. A l'annex B.4, es pot veure l'script que hem fet servir per fer les tasques de clústering.

6.1 Extracció i preparació de dades

Fent servir el mateix mètode d'extracció de dades explicat abans, hem extret les següents dades:

- 3938 tweets que contenen paraules negatives en català o castellà. Les paraules negatives emprades han estat: 'dolent', 'mal', 'horrorós', 'horroroso', 'fatal', 'desastre'.
- 5015 tweets que contenen paraules positives en català o castellà. Les paraules positives emprades han estat: 'bo', 'genial', 'perfecte', 'fantàstic', 'fantàstico', 'perfecto'.

A continuació les dades s'han preparat per poder tractar-les. En concret, hem fet els següents passos.

- Convertir totes les majúscules en minúscules.
- Eliminar els espais blancs sobrers, els números, la puntuació, els accents, els enllaços web i les paraules buides.
- Binaritzar les dades de manera que obtinguem una matriu que ens digui quines paraules aparèixen en cada tweet.

Com hem comentat abans aquests passos de preparació es poden dur a terme amb el package de R *tm* (Feinerer, 2012).

Fent això obtenim una matriu de 20.907 paraules distribuïdes en 8953 documents. De les 187.103.771 cel·les d'aquesta matriu, només 76.600 són diferent de 0. És a dir, com és esperable, la gran majoria de les cel·les tenen valor 0. Finalment, també amb el package *tm*, eliminem els termes dels quals disposem menys dades, i així reduïm la "sparseness" de la matriu.

Si eliminem els mots que hem fet servir per extreure les dades, la figura 6.1 mostra les paraules més freqüents en aquests 8953 tweets.

Podem observar que hi apareixen algunes paraules buides que no s'han eliminat perquè no estaven escrites de manera correcta: per exemple, *qu* o *q*. De la resta destaquen que hi ha força paraules relacionades amb el món del futbol: per exemple, 'Neymar', 'Messi', 'jugar', 'xavi', 'mundodeportivo'.

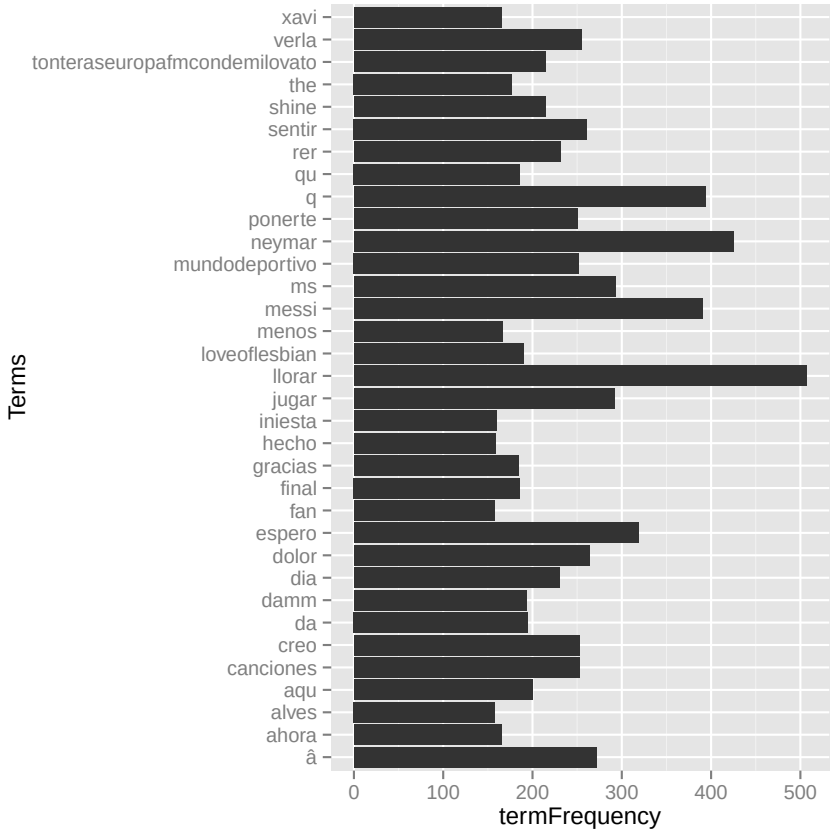

Figure 6.1: *Paraules més freqüents*

6.2 Clusterització de paraules

En un primer moment, hem provat un mètode de clusterització jeràrquica per veure si podíem obtenir relacions interessants entre les paraules. Hem triat el mètode de Ward que hem comentat abans i hem partit de sis clústers inicials. Podem observar la jerarquia de clústers resultant a la figura 6.2.

Bàsicament podem observar que els resultats mostren que les paraules positives tenen tendència a agrupar-se en primer lloc en la jerarquia, mentre que les paraules negatives s'agrupen més cap al final. Per exemple, els dos primers clústers que s'agreguen són els que contenen les paraules "fantàstic"[1] i "perfecte". En canvi, les paraules negatives tenen tendència a agrupar-se cap al final del procés. Hi ha tanmateix dues excepcions: (i) "genial", que és una paraula positiva, s'agrupa al final de tot, just després de dues paraules negatives i (ii) "desastre", que és una paraula negativa, s'ajunta al principi del procés, en tercer lloc, amb les dues primeres paraules positives que s'han agrupat.

[1]La paraula "fantàstic" apareix dos cops ja que la trobem a les dades escrita amb accent i sense.

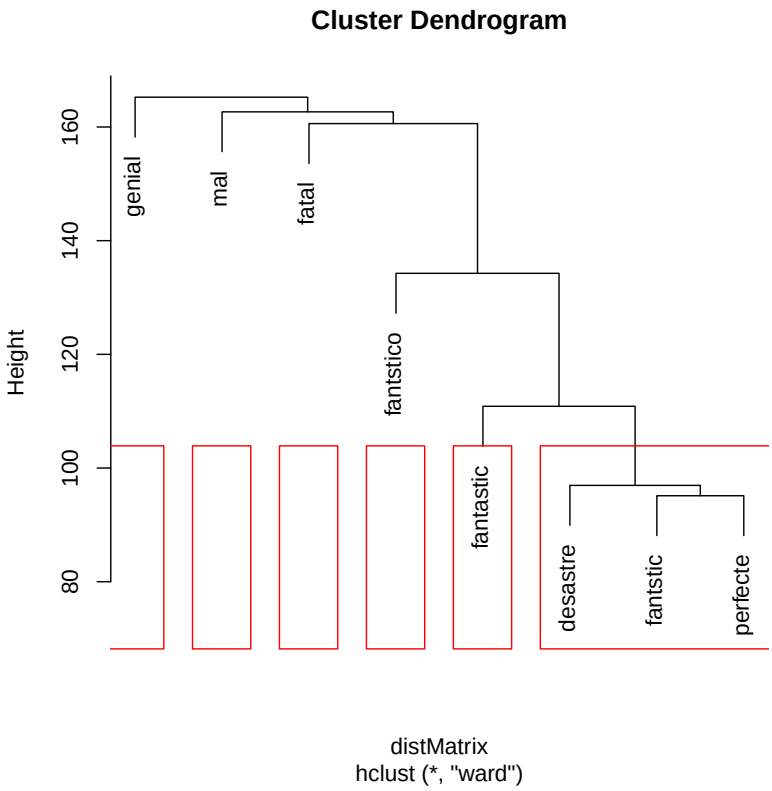

Figure 6.2: *Jerarquia de clústers*

En conclusió, el mètode de clusterització jeràrquica de paraules ofereix alguns resultats prometedors. En la jerarquia podem observar certes tendències de les paraules positives d'agrupar-se primer que les negatives, tendència que es podria emprar per fer altres models de mineria més sofisticats. També és veritat que la jerarquia obtinguda presenta alguns problemes i que caldria veure com es podrien solucionar.

Un dels grans problemes que ens trobem en treballar amb dades lingüístiques és que, com hi ha tanta dispersió, les dades són escasses, com ja havíem comentat en l'apartat d'agregació. És possible que si poguéssim treballar amb un nombre major de dades, la jeràrquica que poguéssim obtenir seria més fiable i més amplia.

6.3 Clusterització de tweets amb l'algoritme *k*-means

En aquest apartat no mirarem de trobar relacions entre paraules, sinó de fer una tasca clàssica de clústering: mirarem de dividir els tweets en diferents particions. Farem servir l'algoritme k-means, que hem comentat abans breument, i el nostre objectiu serà examinar quins resultats obtenim si dividim les dades en 2, 3 i 4 clústers.

Si intentem fer 2 clústers, i mirem quines són les principals paraules que hi apareixen obtenim els següents resultats. Els resultats són força prometedors ja que ens agrupa correctament els tweets que contenen les paraules "genial" i "fantàstic" en el clúster 1 i els tweets que contenen les paraules "mal" i "fatal" en el clúster 2. És a dir, que obtenim un clúster definit principalment per paraules negatives i un altre per paraules positives, tal i com observem a 6.1.

Marca	**Puntuació**
1	genial, fantàstic
2	mal, fatal

Table 6.1: *Dos clusters: paraules més freqüents*

La distribució de tweets positius i negatius en dos clústers es pot veure de manera més gràfica a la figura 6.3. Podem observar que el clúster 2 conté només tweets negatius, mentre que el clúster 1 conté tots els positius, però també un nombre elevat de negatius. Per tant, caldrà veure si fent més clústers podem evitar aquesta barreja de positius i negatius en el mateix clúster.

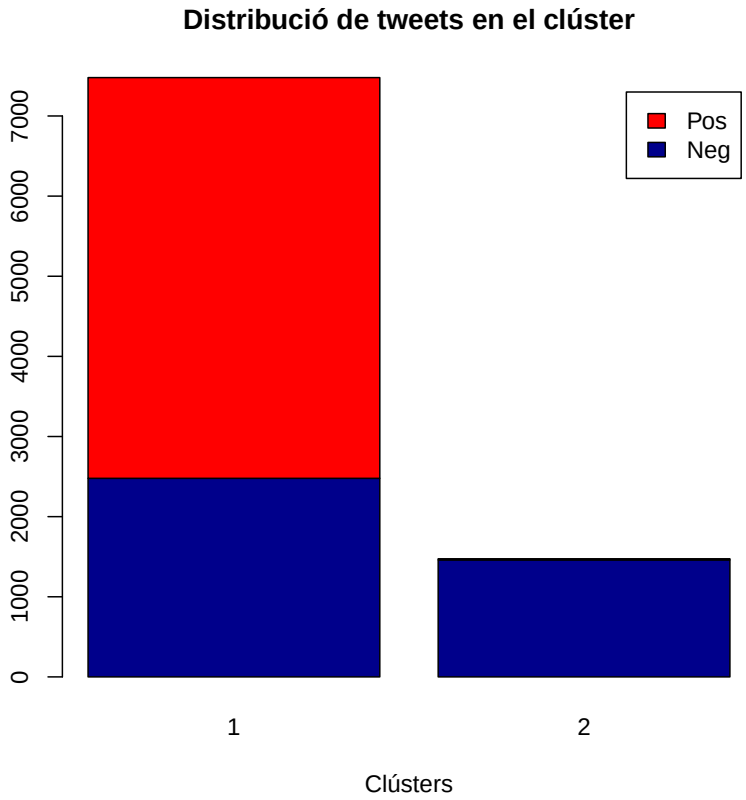

Figure 6.3: *Distribució de tweets positius i negatius en dos clústers*

Si intentem fer 3 clústers, i mirem quines són les principals paraules que hi apareixen obtenim els resultats que mostrem a 6.2. Podem observar que el clúster 1 agrupa paraules

negatives, mentre que el clúster 3 agrupa paraules positives. En canvi, en el clúster 2 hi trobem una barreja, amb paraules com "mal" i "fantástico".

Marca	Puntuació
1	fatal, mal
2	mal, fantástico
3	genial, fantàstic

Table 6.2: *Tres clusters: paraules més freqüents*

La distribució de tweets positius i negatius en dos clústers es pot veure de manera més gràfica a la figura 6.4. Podem observar que el clúster 1 conté només tweets negatius, el clúster 3 conté només tweets positius, mentre que el clúster 2 conté una barreja. Per tant, ara hem aconseguit crear dos clústers "purs", però seguim tenint un clúster (de fet, el que conté més elements) amb tweets de les dues orientacions.

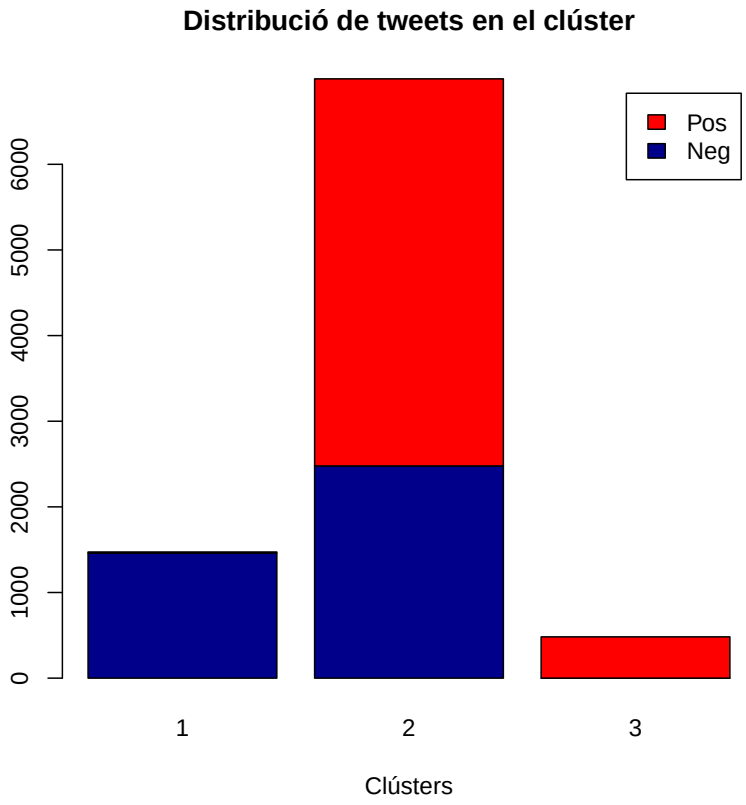

Figure 6.4: *Distribució de tweets positius i negatius en tres clústers*

Si intentem fer 4 clústers, podem veure quines són les principals paraules que hi apareixen a 6.3. Els resultats són força interessants en el sentit que trobem 2 clústers amb paraules clarament negatives i 2 clústers amb paraules clarament positives.

La distribució de tweets positius i negatius en dos clústers es pot veure de manera més

Marca	Puntuació
1	fatal, mal
2	fantástico, fantastic
3	mal, deastre
4	genial, fantàstic

Table 6.3: *Quatre clusters: paraules més freqüents*

gràfica a la figura 6.5. Podem observar que els clúster 1 i 3 contenen només tweets negatius i el clúster 4 només tweets positius. En el clúster 2, trobem una certa barreja, però la proporció de tweets positius és clarament superior a la de tweets negatius.

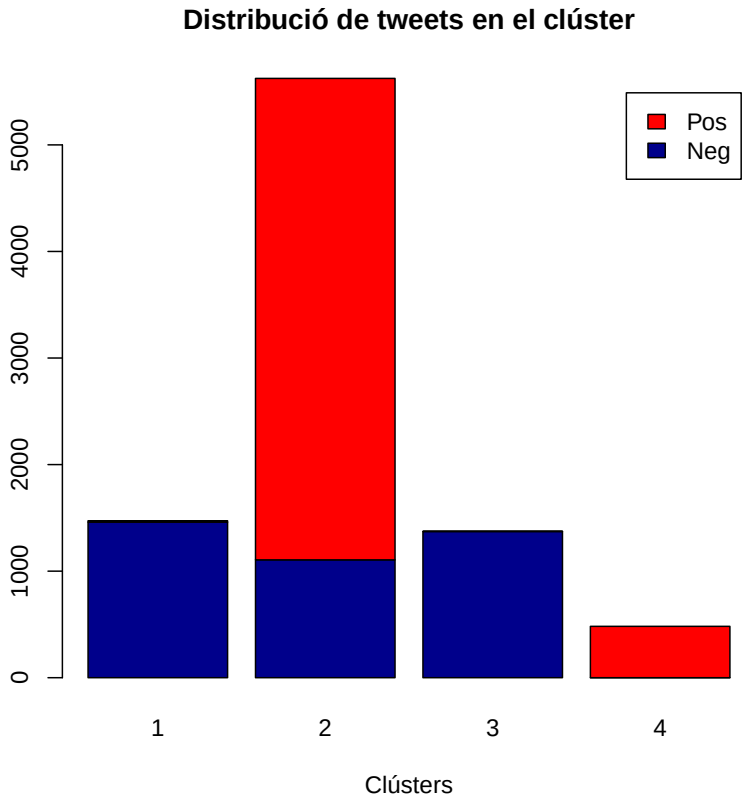

Figure 6.5: *Distribució de tweets positius i negatius en quatre clústers*

Si fem més particions, els resultats ja no milloren. Per tant, el millor model que hem obtingut és el que conté 4 clústers. El model té una taxa d'encert alt, en el sentit que tenim 3 clústers on totes les dades són d'un sol tipus, i tenim un quart clúster en què la majoria de dades són tweets positius, però també n'hi ha de negatius.

En resum, un model basat en clústering es capaç de distingir els tweets positius i negatius que contenen paraules positives i negatives. Dit d'altra manera, és capaç de determinar que un tweet amb la paraula 'mal' té més coses en comú amb un tweet amb la paraula 'fatal' que amb

la paraula ‘genial’. A partir d’aquí, caldria considerar com es pot aprofitar aquesta informació per aplicar-la més directament a un objectiu de negoci.

7
Conclusions

En aquest TFC, he dut a terme algunes aproximacions a la mineria d'opinions a partir de textos de Twitter. En primer lloc, he intentat extreure de les dades opinions sobre certs conceptes (marques de tecnologia, de supermercats, o temes d'interès socials). Per aquest primer objectiu, he provat dues tècniques diferents: un lexicó simple i regles d'associació. Les regles d'associació, malauradament, no han donat cap mena de resultat positiu. En canvi, la tècnica del lexicó simple sí que ens ha donat certa informació, tot i que amb una taxa d'error força alta. A més, els resultats semblen millors per temes de marques comercials que per copsar l'opinió de la societat sobre un tema d'interès social.

Aquests primers intents han mostrar la complexitat de tracta computacionalment dades lingüístiques. D'una banda, les dades lingüístiques solen ser molt disperses: és a dir, hi ha moltes paraules diferents que poden fer servir en un determinat moment i això dificulta que puguem establir relacions entre elles. A més, les dades lingüístiques estan relacionades enter sí de manera força complexes. Per exemple, l'oració (1-a) codifica una opinió clarament positiva. En canvi, l'oració (1-b), tot i contenir pràcticament les mateixes paraules, codifica l'opinió contrària, ja que òbviament la partícula negativa fa canviar l'orientació d'una frase (tot i que no sempre el marcador *no* provoca aquest canvi d'orientació com veiem a (1-c)). Les coses són encara més complicades, ja que com veiem a (1-d), només canviant el temps del verb podem donar a entendre (o en termes més tècnics, implicar) que la opinió actual sobre Apple del qui diu o escriu la frase és negativa.

(1) a. M'agrada Apple.
b. No m'agrada Apple.
c. No només m'agrada Apple, sinó que m'encanta.
d. M'agradava Apple.

En vista de la dificultat de la tasca, en l'últim capítol del treball he decidit per dur a terme una tasca diferent. He fet servir la tècnica del clústering amb l'objectiu de veure si podríem agrupar tots els tweets amb una mateixa orientació (positiva o negativa) i separar-les dels tweets amb orientació contrària. Els resultats d'aquestes proves són encoratjadors en el sentit que

l'algoritme és capaç d'agrupar correctament la majoria de tweets; és a dir, que ha detectat que un tweet positiu t'assembla més a un altre tweet positiu que a un tweet negatiu i viceversa.

Per treball futurs, seria interessant ampliar aquesta mena de treball, agafant més dades (més tweets per paraula positiva i major nombre de paraules positives), i mirar d'extreure automàticament quins trets tenen en comú els textos que són positius o negatius.

Per acabar, voldria també comentar que l'extracció i preparació de dades ha sigut una de les fases del treball que més feina i temps ha necessitat, com és habitual en els projectes de mineria de dades.

Pel que fa a l'extracció de dades, a causa de les restriccions que imposa Twitter, no he pogut treballar amb tantes dades com hauria volgut. Per exemple, una de les opcions que el consultor, el Ramon Caihuelas, em va comentar era fer un seguiment de l'evolució en el temps d'algun tema (una empresa, el fitxatge d'un futbolista, etc). Malauradament, Twitter limita tant el nombre de tweets que es poden extreure com la seva antiguitat, cosa que no és gaire sorprenent tenint en compte de Twitter ven aquestes dades a canvi de xifres astronòmiques[1]. Això ha provocat que segurament no hagi pogut aprofitar tot el potencial de les tècniques de mineria que he utilitzat: per exemple, és probable que amb un volum de dades molt més gran i eliminant els tweets repetits, les regles d'associació haurien donat un resultat millor.

Pel que fa a la preparació de dades, he hagut de fer diverses tasques: netejar les dades de caràcters problemàtics i enllaços, eliminar números, majúscules, signes de puntuació, paraules buides, etc. Cal destacar que els packages de R m'han facilitat molt aquesta mena de tasques que altrament m'haurien demanat encara més temps del que he necessitat.

En resum, aquest treball ha estat una aproximació a la mineria de dades extretes de Twitter, un tema del qual segur que en sentirem a parlar ja que proposa reptes interessants a mig camí entre la informàtica, la privacitat, els interessos comercials i l'extracció de coneixement.

[1] Vegeu, per exemple els articles 'Twitter To Sell Your Old Tweets' o 'Privacy betrayed: Twitter sells multi-billion tweet archive'

A
Regles d'associació

Mercadona

Regla	support	confidence	lift
1 deberea ⇒ eurovision	0.20	1.0000000	4.000000
2 eurovision ⇒ deberea	0.20	0.8000000	4.000000
3 deberea ⇒ espaa	0.20	1.0000000	4.166667
4 espaa ⇒ deberea	0.20	0.8333333	4.166667
5 deberea ⇒ cancion	0.20	1.0000000	3.703704
6 cancion ⇒ deberea	0.20	0.7407407	3.703704
7 eurovision ⇒ espaa	0.20	0.8000000	3.333333
8 espaa ⇒ eurovision	0.20	0.8333333	3.333333
9 eurovision ⇒ cancion	0.24	0.9600000	3.555556
10 cancion ⇒ eurovision	0.24	0.8888889	3.555556
11 espaa ⇒ cancion	0.20	0.8333333	3.086420
12 cancion ⇒ espaa	0.20	0.7407407	3.086420
13 deberea, eurovision ⇒ espaa	0.20	1.0000000	4.166667
14 deberea, espaa ⇒ eurovision	0.20	1.0000000	4.000000
15 espaa, eurovision ⇒ deberea	0.20	1.0000000	5.000000
16 deberea, eurovision ⇒ cancion	0.20	1.0000000	3.703704
17 cancion, deberea ⇒ eurovision	0.20	1.0000000	4.000000
18 cancion, eurovision ⇒ deberea	0.20	0.8333333	4.166667
19 deberea, espaa ⇒ cancion	0.20	1.0000000	3.703704
20 cancion, deberea ⇒ espaa	0.20	1.0000000	4.166667
21 cancion, espaa ⇒ deberea	0.20	1.0000000	5.000000
22 espaa, eurovision ⇒ cancion	0.20	1.0000000	3.703704
23 cancion, eurovision ⇒ espaa	0.20	0.8333333	3.472222
24 cancion, espaa ⇒ eurovision	0.20	1.0000000	4.000000
25 eurovision, mercadona ⇒ cancion	0.22	0.9565217	3.542673
26 cancion, mercadona ⇒ eurovision	0.22	0.8800000	3.520000
27 deberea, espaa, eurovision ⇒ cancion	0.20	1.0000000	3.703704
28 cancion, deberea, eurovision ⇒ espaa	0.20	1.0000000	4.166667
29 cancion, deberea, espaa ⇒ eurovision	0.20	1.0000000	4.000000
30 cancion, espaa, eurovision ⇒ deberea	0.20	1.0000000	5.000000

Carrefour

Regla	support	confidence	lift
1 indiescabreados ⇒ director	0.20	1.0000000	4.761905
2 director ⇒ indiescabreados	0.20	0.9523810	4.761905
3 indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
4 affleck ⇒ indiescabreados	0.20	0.9523810	4.761905
5 indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
6 cajero ⇒ indiescabreados	0.20	0.9090909	4.545455
7 indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
8 mejor ⇒ indiescabreados	0.20	0.7142857	3.571429
9 director ⇒ affleck	0.21	1.0000000	4.761905
10 affleck ⇒ director	0.21	1.0000000	4.761905
11 director ⇒ cajero	0.21	1.0000000	4.545455
12 cajero ⇒ director	0.21	0.9545455	4.545455
13 director ⇒ mejor	0.21	1.0000000	3.571429
14 mejor ⇒ director	0.21	0.7500000	3.571429
15 affleck ⇒ cajero	0.21	1.0000000	4.545455
16 cajero ⇒ affleck	0.21	0.9545455	4.545455
17 affleck ⇒ mejor	0.21	1.0000000	3.571429
18 mejor ⇒ affleck	0.21	0.7500000	3.571429
19 cajero ⇒ mejor	0.21	0.9545455	3.409091
20 mejor ⇒ cajero	0.21	0.7500000	3.409091
21 director, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
22 affleck, indiescabreados ⇒ director	0.20	1.0000000	4.761905
23 affleck, director ⇒ indiescabreados	0.20	0.9523810	4.761905
24 director, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
25 cajero, indiescabreados ⇒ director	0.20	1.0000000	4.761905
26 cajero, director ⇒ indiescabreados	0.20	0.9523810	4.761905
27 director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
28 indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
29 director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
30 carrefour, indiescabreados ⇒ director	0.20	1.0000000	4.761905
31 carrefour, director ⇒ indiescabreados	0.20	0.9523810	4.761905
32 affleck, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
33 cajero, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
34 affleck, cajero ⇒ indiescabreados	0.20	0.9523810	4.761905
35 affleck, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429

36 indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
37 affleck, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
38 carrefour, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
39 affleck, carrefour ⇒ indiescabreados	0.20	0.9523810	4.761905
40 cajero, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
41 indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
42 cajero, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
43 carrefour, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
44 cajero, carrefour ⇒ indiescabreados	0.20	0.9090909	4.545455
45 carrefour, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
46 carrefour, mejor ⇒ indiescabreados	0.20	0.7142857	3.571429
47 affleck, director ⇒ cajero	0.21	1.0000000	4.545455
48 cajero, director ⇒ affleck	0.21	1.0000000	4.761905
49 affleck, cajero ⇒ director	0.21	1.0000000	4.761905
50 affleck, director ⇒ mejor	0.21	1.0000000	3.571429
51 director, mejor ⇒ affleck	0.21	1.0000000	4.761905
52 affleck, mejor ⇒ director	0.21	1.0000000	4.761905
53 carrefour, director ⇒ affleck	0.21	1.0000000	4.761905
54 affleck, carrefour ⇒ director	0.21	1.0000000	4.761905
55 cajero, director ⇒ mejor	0.21	1.0000000	3.571429
56 director, mejor ⇒ cajero	0.21	1.0000000	4.545455
57 cajero, mejor ⇒ director	0.21	1.0000000	4.761905
58 carrefour, director ⇒ cajero	0.21	1.0000000	4.545455
59 cajero, carrefour ⇒ director	0.21	0.9545455	4.545455
60 carrefour, director ⇒ mejor	0.21	1.0000000	3.571429
61 carrefour, mejor ⇒ director	0.21	0.7500000	3.571429
62 affleck, cajero ⇒ mejor	0.21	1.0000000	3.571429
63 affleck, mejor ⇒ cajero	0.21	1.0000000	4.545455
64 cajero, mejor ⇒ affleck	0.21	1.0000000	4.761905
65 affleck, carrefour ⇒ cajero	0.21	1.0000000	4.545455
66 cajero, carrefour ⇒ affleck	0.21	0.9545455	4.545455
67 affleck, carrefour ⇒ mejor	0.21	1.0000000	3.571429
68 carrefour, mejor ⇒ affleck	0.21	0.7500000	3.571429
69 cajero, carrefour ⇒ mejor	0.21	0.9545455	3.409091
70 carrefour, mejor ⇒ cajero	0.21	0.7500000	3.409091
71 affleck, director, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
72 cajero, director, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
73 affleck, cajero, indiescabreados ⇒ director	0.20	1.0000000	4.761905
74 affleck, cajero, director ⇒ indiescabreados	0.20	0.9523810	4.761905
75 affleck, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
76 director, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
77 affleck, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905

78 affleck, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
79 carrefour, director, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
80 affleck, carrefour, indiescabreados ⇒ director	0.20	1.0000000	4.761905
81 affleck, carrefour, director ⇒ indiescabreados	0.20	0.9523810	4.761905
82 cajero, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
83 director, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
84 cajero, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
85 cajero, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
86 carrefour, director, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
87 cajero, carrefour, indiescabreados ⇒ director	0.20	1.0000000	4.761905
88 cajero, carrefour, director ⇒ indiescabreados	0.20	0.9523810	4.761905
89 carrefour, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
90 carrefour, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
91 carrefour, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
92 affleck, cajero, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
93 affleck, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
94 cajero, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
95 affleck, cajero, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
96 affleck, carrefour, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
97 cajero, carrefour, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
98 affleck, cajero, carrefour ⇒ indiescabreados	0.20	0.9523810	4.761905
99 affleck, carrefour, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
100 carrefour, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
101 affleck, carrefour, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
102 cajero, carrefour, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
103 carrefour, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
104 cajero, carrefour, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
105 affleck, cajero, director ⇒ mejor	0.21	1.0000000	3.571429
106 affleck, director, mejor ⇒ cajero	0.21	1.0000000	4.545455
107 cajero, director, mejor ⇒ affleck	0.21	1.0000000	4.761905
108 affleck, cajero, mejor ⇒ director	0.21	1.0000000	4.761905
109 affleck, carrefour, director ⇒ cajero	0.21	1.0000000	4.545455
110 cajero, carrefour, director ⇒ affleck	0.21	1.0000000	4.761905
111 affleck, cajero, carrefour ⇒ director	0.21	1.0000000	4.761905
112 affleck, carrefour, director ⇒ mejor	0.21	1.0000000	3.571429
113 carrefour, director, mejor ⇒ affleck	0.21	1.0000000	4.761905
114 affleck, carrefour, mejor ⇒ director	0.21	1.0000000	4.761905
115 cajero, carrefour, director ⇒ mejor	0.21	1.0000000	3.571429
116 carrefour, director, mejor ⇒ cajero	0.21	1.0000000	4.545455
117 cajero, carrefour, mejor ⇒ director	0.21	1.0000000	4.761905
118 affleck, cajero, carrefour ⇒ mejor	0.21	1.0000000	3.571429
119 affleck, carrefour, mejor ⇒ cajero	0.21	1.0000000	4.545455

120 cajero, carrefour, mejor ⇒ affleck	0.21	1.0000000	4.761905
121 affleck, cajero, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
122 affleck, director, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
123 cajero, director, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
124 affleck, cajero, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
125 affleck, cajero, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
126 affleck, carrefour, director, indiescabreados ⇒ cajero	0.20	1.0000000	4.545455
127 cajero, carrefour, director, indiescabreados ⇒ affleck	0.20	1.0000000	4.761905
128 affleck, cajero, carrefour, indiescabreados ⇒ director	0.20	1.0000000	4.761905
129 affleck, cajero, carrefour, director ⇒ indiescabreados	0.20	0.9523810	4.761905
130 affleck, carrefour, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
131 carrefour, director, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
132 affleck, carrefour, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
133 affleck, carrefour, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
134 cajero, carrefour, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
135 carrefour, director, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
136 cajero, carrefour, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
137 cajero, carrefour, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
138 affleck, cajero, carrefour, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
139 affleck, carrefour, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
140 cajero, carrefour, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
141 affleck, cajero, carrefour, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905
142 affleck, cajero, carrefour, director ⇒ mejor	0.21	1.0000000	3.571429
143 affleck, carrefour, director, mejor ⇒ cajero	0.21	1.0000000	4.545455
144 cajero, carrefour, director, mejor ⇒ affleck	0.21	1.0000000	4.761905
145 affleck, cajero, carrefour, mejor ⇒ director	0.21	1.0000000	4.761905
146 affleck, cajero, carrefour, director, indiescabreados ⇒ mejor	0.20	1.0000000	3.571429
147 affleck, carrefour, director, indiescabreados, mejor ⇒ cajero	0.20	1.0000000	4.545455
148 cajero, carrefour, director, indiescabreados, mejor ⇒ affleck	0.20	1.0000000	4.761905
149 affleck, cajero, carrefour, indiescabreados, mejor ⇒ director	0.20	1.0000000	4.761905
150 affleck, cajero, carrefour, director, mejor ⇒ indiescabreados	0.20	0.9523810	4.761905

Iberia: Vegeu la pàgina web: http://laiamayol.wordpress.com/tfc/

Escrache: Vegeu la pàgina web: http://laiamayol.wordpress.com/tfc/

Colau: Vegeu la pàgina web: http://laiamayol.wordpress.com/tfc/

B
Scripts de R

B.1 Script per baixar dades de Twitter

```
# Script per baixar-nos dades de Twitter
library(twitteR)
library(ROAuth)
library(plyr)

## Establir el directori
setwd("C:\\TFC")

download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

### 1. Autenticar-nos  a Twitter

## Cal haver registrat una app i tenir una Key i un password.
KEY <- "yourKey"
SECRET <- "yourSecret"

## Creem un objecte per autenticar-nos
cred <- OAuthFactory$new(consumerKey = KEY,
    consumerSecret = SECRET,
    requestURL = "https://api.twitter.com/oauth/request_token",
    accessURL = "https://api.twitter.com/oauth/access_token",
    authURL = "https://api.twitter.com/oauth/authorize")
cred$handshake(cainfo="cacert.pem")

## Salvem l'objecte Cred
save(cred, file="twitter authentication.Rdata")

## Autentiquem-nos
registerTwitterOAuth(cred)


## 2. Extreure dades
```

```
gsub2 <- function(pattern, replacement, x, ...) {
for(i in 1:length(pattern))
x <- gsub(pattern[i], replacement[i], x, ...)
x
}

# Cerquem una paraula, extraiem el text, traiem els caràcters problemàtics.

PAH = searchTwitter('PAH', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
PAH.text = laply(PAH, function(t) t$getText() )
PAH.text <- gsub2(from, to, PAH.text)
PAH.text <- enc2utf8(gsub("[\U80-\UFF]", "", PAH.text, useBytes=TRUE))

Escrache = searchTwitter('escrache', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Escrache.text = laply(Escrache, function(t) t$getText() )
Escrache.text <- gsub2(from, to, Escrache.text)
Escrache.text <- enc2utf8(gsub("[\U80-\UFF]", "", Escrache.text, useBytes=TRUE))

Colau = searchTwitter('Colau', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Colau.text = laply(Colau, function(t) t$getText() )
Colau.text <- gsub2(from, to, Colau.text)
Colau.text <- enc2utf8(gsub("[\U80-\UFF]", "", Colau.text, useBytes=TRUE))

Apple = searchTwitter('Apple', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Apple.text = laply(Apple , function(t) t$getText() )
Apple.text <- gsub2(from, to, Apple.text)
Apple.text <- enc2utf8(gsub("[\U80-\UFF]", "", Apple.text, useBytes=TRUE))

Samsung = searchTwitter('Samsung', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Samsung.text = laply(Samsung, function(t) t$getText() )
Samsung.text <- gsub2(from, to, Samsung.text)
Samsung.text <- enc2utf8(gsub("[\U80-\UFF]", "", Samsung.text, useBytes=TRUE))

Iberia = searchTwitter('Iberia', n=1500, geocode='42.18221,2.48802,40mi',
```

```
 cainfo="cacert.pem")
Iberia.text = laply(Iberia, function(t) t$getText() )
Iberia.text <- gsub2(from, to, Iberia.text)
Iberia.text <- enc2utf8(gsub("[\U80-\UFF]", "", Iberia.text, useBytes=TRUE))

Caprabo = searchTwitter('Caprabo', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Caprabo.text = laply(Caprabo, function(t) t$getText() )
Caprabo.text <- gsub2(from, to, Caprabo.text)
Caprabo.text <- enc2utf8(gsub("[\U80-\UFF]", "", Caprabo.text, useBytes=TRUE))

Mercadona = searchTwitter('Mercadona', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Mercadona.text = laply(Mercadona, function(t) t$getText() )
 Mercadona.text <- gsub2(from, to, Mercadona.text)
Mercadona.text <- enc2utf8(gsub("[\U80-\UFF]", "", Mercadona.text, useBytes=TRUE)

Carrefour = searchTwitter('Carrefour', n=1500, geocode='42.18221,2.48802,40mi',
cainfo="cacert.pem")
Carrefour.text = laply(Carrefour, function(t) t$getText() )
Carrefour.text <- gsub2(from, to, Carrefour.text)
Carrefour.text <- enc2utf8(gsub("[\U80-\UFF]", "", Carrefour.text, useBytes=TRUE)
```

B.2 Script per fer mineria amb un lexicó simple

```
## Carreguem els arxius amb les paraules positives i negatives
neg.words = scan('sol_negativas.txt', what='character', comment.char=';')
pos.words = scan('sol_positivas.txt', what='character', comment.char=';')

## Definim una funció que calculi la valoració de cada Tweet
score.sentiment = function(sentences, pos.words, neg.words, .progress='none')
{
    require(plyr)
    require(stringr)

    scores = laply(sentences, function(sentence, pos.words, neg.words) {

        # Netegem les frases
        sentence = gsub('[[:punct:]]', '', sentence)
        sentence = gsub('[[:cntrl:]]', '', sentence)
        sentence = gsub('\\d+', '', sentence)
        # i posem en minúscules
        sentence = tolower(sentence)

        # dividim les frases en paraules
        word.list = str_split(sentence, '\\s+')
        words = unlist(word.list)

        # comparem les paraules amb les paraules del nostre lexicó.
        pos.matches = match(words, pos.words)
        neg.matches = match(words, neg.words)
        pos.matches = !is.na(pos.matches)
        neg.matches = !is.na(neg.matches)

        # agreguem totes les puntuacions per cada tweet
        score = sum(pos.matches) - sum(neg.matches)

        return(score)
    }, pos.words, neg.words, .progress=.progress )

    scores.df = data.frame(score=scores, text=sentences)
    return(scores.df)
}
```

```
## Computem les mitjanes per cada conjunt de dades i fem l'histograma.

Iberia.scores = score.sentiment(Iberia.text, pos.words, neg.words)
hist(iberia.scores$score)

Apple.scores = score.sentiment(Apple.text, pos.words, neg.words)
hist(Apple.scores$score)

Samsung.scores = score.sentiment(Samsung.text, pos.words, neg.words)
hist(Samsung.scores$score)

Escrache.scores = score.sentiment(Escrache.text, pos.words, neg.words)
hist(Escrache.scores$score)

PAH.scores = score.sentiment(PAH.text, pos.words, neg.words)
hist(PAH.scores$score)

Colau.scores = score.sentiment(Colau.text, pos.words, neg.words)
hist(Colau.scores$score)

Carrefour.scores = score.sentiment(Carrefour.text, pos.words, neg.words)
hist(Carrefour.scores$score)

Caprabo.scores = score.sentiment(Caprabo.text, pos.words, neg.words)
hist(Caprabo.scores$score)

Mercadona.scores = score.sentiment(Mercadona.text, pos.words, neg.words)
hist(Mercadona.scores$score)
```

B.3 Script per fer mineria amb regles d'associació

```
## Preparem les dades
require(tm)
require(arules)

# 'data' ha de ser el fitxer de dades que volguem examinar.
test <- Corpus(VectorSource(data))
test  <- tm_map(test, stripWhitespace)
test  <- tm_map(test, removeNumbers)
test  <- tm_map(test, removePunctuation)
test  <- tm_map(test, tolower)
removeURL <- function(x) gsub("http[[:alnum:]]*", "", x)
test  <- tm_map(test, removeURL)
test  <- tm_map(test, removeWords,  c("RT", "rt", stopwords("spanish"),
stopwords("catalan")))

## fem la matriu de dades
test <- DocumentTermMatrix(test)
test <- TermDocumentMatrix(test, control=list(wordLengths=c(1,Inf)))
mat <- as.matrix(test)

## mirem les regles d'associació
rules <- apriori(mat, parameter = list(sup = 0.2, conf = 0.5,target="rules"))
inspect(rules)
```

B.4 Script per fer mineria amb clústering

```
library(plyr)
# Ajuntem tots els fitxers que contenen paraules positives
pos.data <- c(bo.text, genial.text, perfecte.text, fantastic.text,
perfecto.text, fantastico.text)
pos.df = data.frame(pos.data)
pos.df$tipus= 'Pos'
colnames(pos.df) <- c("text","tipus")

# Ajuntem tots els fitxers que contenen paraules negatives
neg.data <- c(dolent.text, horroros.text, fatal.text, desastre.text,
horroroso.text, mal.text)
neg.df = data.frame(neg.data)
neg.df$tipus= 'Neg'
colnames(neg.df) <- c("text","tipus")

# Ajuntem les dades i les preparem
total.data <- c(neg.df, pos.df)
test <- Corpus(VectorSource(total.data$text))
test  <- tm_map(test, stripWhitespace)
test  <- tm_map(test, removeNumbers)
test  <- tm_map(test, removePunctuation)
test  <- tm_map(test, tolower)
removeURL <- function(x) gsub("http[[:alnum:]]*", "", x)
test  <- tm_map(test, removeURL)

myStopWords <- c("RT", "rt", stopwords("spanish"), stopwords("catalan"))
myStopWords <- setdiff(myStopWords, c("bo", "mal", "malament", "genial",
 "horroros", "horroroso", "perfecte", "perfecto", "fatal", "fantastico",
"fantastic", "fantstic", "fantstico", "malo", "dolent", "desastre"))
test  <- tm_map(test, removeWords,  myStopWords)
test<- TermDocumentMatrix(test, control=list(wordLengths=c(1,Inf)))

# Fem el gràfic de les paraules més freqüents
termFrequency <- rowSums(as.matrix(test))
termFrequency <- subset(termFrequency, termFrequency>=150)
library(ggplot2)
qplot(names(termFrequency), termFrequency, geom="bar", xlab="Terms") +
coord_flip()
```

```
#Fem clustering de paraules

myTdm2 <- removeSparseTerms(test, sparse=0.95)
m2 <- as.matrix(myTdm2)

distMatrix <- dist(scale(m2))
fit <- hclust(distMatrix, method="ward")
plot(fit)
rect.hclust(fit, k=3)
(groups <- cutree(fit, k=3))

#Fem clustering de tweets amb k-means
 # transposem la matriu
 m3 <- t(m2)

 # apliquem el k-means
 k <- 4
 kmeansResult <- kmeans(m3, k)
 round(kmeansResult$centers, digits=3)

# Cerquem les paraules més freqüents a cada clúster
 for (i in 1:k) {
 cat(paste("cluster ", i, ": ", sep=""))
 s <- sort(kmeansResult$centers[i,], decreasing=T)
 cat(names(s)[1:2], "\n")
 }

# Fem els gràfics per veure la distribució de tweets en el clúster
cluster <- data.frame(kmeansResult$cluster)
x <- data.frame(id=1:8953)
cluster["ID"] <- x

results <- merge(total.data, cluster)
attach(results)

counts <- table(tipus, kmeansResult$cluster)
barplot(counts, main="Distribució de tweets en el clúster",
  xlab="Clústers", col=c("darkblue","red"),
          legend = rownames(counts))
```

Bibliografia

Breen, J. (2011). R by example: mining twitter for consumer attitudes towards airlines. A *Boston Predictive Analytics MeetUp*. http://jeffreybreen.wordpress.com/2011/07/04/twitter-text-mining-r-slides/.

Feinerer, I. (2012). tm: Text mining package. *R package version 0.5-7.1*.

Gentry, J. (2013). *twitteR: R based Twitter client*. R package version 1.1.0.

Hahsler, M., Grün, B., i Hornik, K. (2007). arules: Mining association rules and frequent itemsets, 2006, url http://cran. r-project. org/, r package version. A *SIGKDD Explorations*. Citeseer.

Hu, M. i Liu, B. (2004). Mining and summarizing customer reviews. A *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pàgines 168–177. ACM.

Liu, B. (2010). Sentiment analysis and subjectivity. *Handbook of natural language processing*, *2*, 568.

Martínez-Cámara, E., Martín-Valdivia, M. T., Molina-González, M. D., i Ureña-López, L. A. (2013). Bilingual experiments on an opinion comparable corpus. A *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pàgines 87–93. ACL.

Pang, B. i Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. A *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pàgines 115–124. Association for Computational Linguistics.

Pang, B., Lee, L., i Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. A *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pàgines 79–86. Association for Computational Linguistics.

RDevelopment, C. (2011). Team. 2008. r: A language and environment for statistical computing. r foundation for statistical computing, vienna, austria. *ISBN*, 3–900051.

Sangüesa, R. (2012). Agregacio (clustering). A *Mineria de dades (Material docent de la UOC*.

Sangüesa, R. i i Luis Molina (2012). Regles d'associació. A *Mineria de dades (Material docent de la UOC*.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. A *Proceedings of the 40th annual meeting on association for computational linguistics*, pàgines 417–424. Association for Computational Linguistics.