Com interpretar i analitzar automàticament la informació textual

PID_00257760

Joaquim Moré

Temps mínim de dedicació recomanat: 4 hores

Universitat Oberta de Catalunya

Joaquim Moré

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats pel professor: Jordi Casas (2019)

Primera edició: setembre 2019
Autoria: Joaquim Moré
Llicència CC BY-SA d'aquesta edició, FUOC, 2019
Av. Tibidabo, 39-43, 08035 Barcelona
Realització editorial: FUOC

Índex

Introducció

En aquest mòdul explicarem els fonaments del processament del llenguatge natural (PLN) i les tècniques bàsiques per a processar la informació textual de manera que puguem abordar l'anàlisi d'opinions i sentiments.

Explicarem els fonaments del PLN basant-nos en el cas d'ús que presentem a continuació.

El cas

El cas d'ús està inspirat en les dades d'un *kernel* que té per nom *March 2018 NYT Headline Click Bait*, de la competició Kaggle New York Times Comments. Està disponible en l'enllaç https://www.kaggle.com/aashita/nyt-comments.

Els directius de la versió digital del *New York Times* decideixen posar publicitat dels seus patrocinadors al costat de les notícies que provoquen més comentaris.

Per això pensen a invertir en la predicció de titulars que motiven comentaris. D'aquesta manera, sabran quines notícies poden anar acompanyades de publicitat.

Seguint les indicacions d'uns consultors, els directius decideixen contractar algú que s'ocupi de descobrir quins continguts dels titulars provoquen més comentaris.

Els consultors s'ofereixen a fer aquest servei per una quantitat elevada. Argumenten que hauria de subcontractar un exèrcit d'especialistes en comunicació i màrqueting per a analitzar a fons els titulars del *New York Times* de manera manual.

Afortunadament, els directius del *New York Times* coneixen una petita empresa anomenada S&S (Sense and Suitability), especialitzada en processament de llenguatge natural. En el passat S&S havia automatitzat alguns processos de cerca d'informació i anàlisi de les notícies per part dels redactors i, per tant, els directius pensen que l'anàlisi dels titulars també es pot automatitzar aplicant tècniques de PLN, la qual cosa comportaria un estalvi de diners important.

Els directius es posen en contacte amb S&S, que es mostra interessada en el projecte. L'empresa planteja una proposta: en un mes tindran preparada una prova de concepte per a demostrar que és viable fer, amb mètodes de PLN, una anàlisi de titulars per a classificar els que són motivadors d'opinió.

L'objectiu d'aquest mòdul és donar resposta a les preguntes metodològiques bàsiques. Són les preguntes que l'equip de S&S ha de plantejar-se i a les quals ha de donar resposta, assumint que els continguts d'un text, sigui un titular o qualsevol altre document, són en realitat les seves paraules. Les preguntes que l'equip de S&S es planteja són les següents:

1) Què considerem paraula?
2) Com es detecten i processen les paraules?
3) Com es poden obtenir les dades que relacionen les paraules d'un text amb les opinions que provoca?
4) Com es pot aprofitar aquesta relació per a predir la reacció dels lectors?
5) Quines eines i recursos hi ha disponibles?

1. Què es considera paraula?

Des del punt de vista del processament del llenguatge natural, la resposta a aquesta pregunta pot semblar òbvia. Donat un text, la paraula és una seqüència de caràcters entre dos espais en blanc: entre l'inici de línia i un espai en blanc o bé entre un espai en blanc i el final de línia. I, si volem obtenir les paraules del text, n'hi ha prou de trencar la seqüència de caràcters (*split* en cas que utilitzem un terme d'ús comú en els llenguatges de programació) allà on hi ha un espai en blanc. D'aquesta manera, podem obtenir les paraules dels titulars del *New York Times*. En la taula 1 podem veure la llista de paraules d'un titular obtinguda segons aquest criteri. En el *notebook* PLA-1 1.1 es mostra com es fa en Python.

Taula 1. Paraules d'un titular segons el mètode de *split*

Titular: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".
['The', '"Daily', 'Show"', 'host', 'said', 'the', 'attacks', 'of', 'the', 'president', 'on', 'fellow', 'Republicans', 'at', 'a', 'meeting', 'reminded', 'him', 'of', '"a', 'drunk', 'uncle', 'calling', 'everyone', 'out', 'at', 'a', 'wedding."']

L'anàlisi dels resultats obtinguts pel mètode de *split* fa que aflorin les diferents maneres que tenen els membres de S&S d'abordar l'encàrrec.

D'una banda, Peter, que és el coordinador de l'equip de desenvolupament, sap per experiència que algorismes molt simples poden solucionar problemes complexos. Ell pensa de provar una anàlisi estrictament quantitativa que consisteix a verificar si la longitud dels titulars està relacionada amb el nombre de comentaris. Joseph, un enginyer informàtic especialista en aprenentatge automàtic, està d'acord a explorar aquest mètode. Per a Peter i Joseph, la llista obtinguda pel mètode de *split* els sembla adequada.

En canvi, Beth, Anne i Henry són uns membres de l'equip que creuen que s'ha d'abordar l'objectiu amb criteris qualitatius, concretament semàntics. Pensen que el contingut semàntic del titular, el que diu, influeix en els lectors i els motiva a escriure comentaris. Amb aquesta idea en ment, Beth, Anne i Henry no estan d'acord a considerar tots els elements de la llista de la taula 1 com a paraules. Vegem algunes de les seves raons.

1.1. La paraula com a unitat de significant i significat

Bibliografia

Sobre la paraula com a símbol compost de significant i significat, vegeu *Cours de linguistique générale*, de **Ferdinand de Saussure**, **Charles Bally** i **Albert Sechehaye** (1916).

«Wedding. "», per exemple, no és una paraula, encara que sí que ho és *wedding*. *Wedding*, sense les cometes i el punt, té un contingut semàntic, mentre que «wedding. "» no el té. Per la mateixa raó, en comptes de «"Daily» s'hauria de considerar *Daily*, en comptes de «Show"» s'hauria de considerar *Show*, i en comptes de «"a» s'hauria de considerar *a*. Per a Beth, que va estudiar filologia i té molt interioritzada la noció de paraula que li van ensenyar a la facultat, les cometes i els signes de puntuació són elements tipogràfics que s'adhereixen a la paraula en els textos escrits però que no tenen presència en altres canals de comunicació. La paraula és un signe compost de significant i significat en el mitjà escrit i també en l'oral. El significant d'una paraula és la forma, que correspon a la seva pronunciació, i el significat és el concepte al qual fa referència la paraula. Per tant, les paraules de la llista haurien d'estar despullades d'elements tipogràfics que trenquen la relació entre el significant i el seu significat.

Per tant, Beth, Anne i Henry volen obtenir una llista en la qual els elements amb signes de puntuació no es distingeixin de l'element sense signe de puntuació. D'aquesta manera, també serà possible explicar la freqüència de paraules que tenen el mateix significat. A continuació es compara la freqüència de les paraules segons el mètode de *split* i la freqüència de paraules segons la relació del significant amb el significat (vegeu el *notebook* PLA-1 1.2).

Taula 2. Freqüències de les paraules del titular segons el mètode de *split* i de relació de significant i significat

Titular: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".	
Mètode de *split*	'at': 2, 'a': 2, 'The': 1, '"Daily': 1, 'Show"': 1, 'host': 1, 'said': 1, 'the': 1, 'president's': 1, 'attacks': 1, 'on': 1, 'fellow': 1, 'Republicans': 1, 'meeting': 1, 'reminded': 1, 'him': 1, 'of': 1, '"a': 1, 'drunk': 1, 'uncle': 1, 'calling': 1, 'everyone': 1, 'out': 1, 'wedding."': 1
Relació de significant amb significat	'a ': 3, 'at ': 2, 'The ': 1, 'Daily': 1, 'Show ': 1, 'host ': 1, 'said ': 1, 'the ': 2, 'attacks': 1, 'president ': 1, 'on': 1, 'fellow': 1, 'Republicans': 1, 'meeting': 1, 'reminded': 1, 'him': 1, 'of': 2, 'drunk': 1, 'uncle': 1, 'calling': 1, 'everyone': 1, 'out': 1, 'wedding': 1

És evident que el fet que *the* estigui en majúscula o minúscula no n'afecta el sentit. Per tant, *The* i *the* es poden explicar com una sola paraula. Si no distingim entre majúscules i minúscules, la freqüència de les paraules (totes en minúscules) queda així:

Taula 3. Freqüències de les paraules del titular sense distingir majúscules i minúscules

Titular: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".

Relació de significant amb significat	'a': 3, 'the': 3, 'at': 2, 'daily': 1, 'xou': 1, 'host ': 1,' said ': 1,' president ': 1,' attacks ': 1,' on ': 1,' fellow ': 1,' republicans ': 1,' meeting ': 1, 'reminded': 1, 'him': 1, 'of': 2, 'drunk': 1, 'uncle': 1, 'calling': 1, 'everyone': 1, 'out': 1, 'wedding': 1

1.2. Termes *monoparaula* i *multiparaula*

Davant la llista de la taula 3, Anne s'adona que s'ha trencat la relació estreta que hi havia entre les paraules *daily* i *show*. Aquestes dues paraules són ara independents, amb el seu significant i significat, quan el redactor havia utilitzat el recurs de posar unes cometes («*Daily Show*») per a indicar al lector que les dues paraules es referien a una sola cosa. Henry fins i tot creu que el redactor havia considerat la combinació *Daily Show host* com la referència a un únic concepte.

Anne, Beth i Henry plantegen la possibilitat de considerar també combinacions de dues o més paraules com una sola unitat de significat, igual que *New York Times*. *New York Times* és una unitat formada per tres paraules que fa referència a un diari i no a *New* ni a la ciutat de *York* ni a *Times* per separat.

Els tres consideren que aquestes combinacions de paraules formen una paraula per si mateixa. Així, doncs, una paraula és també la combinació de dues o més paraules. Això sí, el significat d'aquesta paraula transcendeix o és diferent de la unió dels significats de les paraules que la formen. Aquesta consideració de paraula és diferent de la consideració de paraula com una sola unitat que tenen Peter i Joseph, que és la més popular.

Per a evitar malentesos, Beth, Anne i Henry decideixen adoptar la noció de terme. Un terme pot ser monoparaula o multiparaula. En el camp de la lingüística computacional, els termes multiparaula es denominen també *col·locacions* o *multiwords*.

2. Preprocessar o no preprocessar?

Tal com hem dit, Peter —que és el coordinador— vol explorar la relació entre la longitud dels titulars i el nombre de comentaris, mentre que Beth defensa una aproximació basada en la relació entre el significant i el significat. En la segona reunió d'equip, Peter decideix que Joseph explori la relació entre la longitud dels titulars i el nombre de comentaris i decideix també que, mentre Joseph explora la via quantitativa, Beth, Anne i Henry treballin en la via més semàntica.

L'anàlisi dels titulars per part de Peter no requereix un preprocessament del text, però l'anàlisi que han de fer Beth, Anne i Henry sí —tal com explicarem en aquest apartat.

Peter decideix reunir-se tres dies després per contrastar els resultats obtinguts i decidir quina és la millor aproximació.

2.1. El corpus d'anàlisi

Peter demana al *New York Times* una mostra de titulars per a poder preparar la proposta. L'endemà, el *New York Times* entrega a S&S un full de càlcul que recull tots els titulars del mes de març del 2018 amb els respectius cossos de notícia i el nombre de comentaris.

2.2. Preprocessament del text i les seves eines

Tal com hem dit, el mètode d'obtenció de paraules via *split* ja va bé a Joseph, encara que des del punt de vista gramatical i filològic sigui discutible. No obstant això, per a Beth, Anne i Henry, la detecció de les paraules requereix un processament previ del text. Aquest processament es fa amb les eines que explicarem a continuació a mesura que Beth, Anne i Henry hagin de solucionar les dificultats normals en aquesta tasca.

2.2.1. Tokenizer

El **tokenizer** és l'eina bàsica del processament del llenguatge natural. *Tokenizer* és un terme anglès que és difícil de traduir al català amb una sola paraula. És per això que ens referirem a aquesta eina amb el terme original, que és d'ús comú entre especialistes en PLN.

Donat un text, el *tokenizer* llista els *tokens* d'aquest text. *Token* és un altre terme anglès d'ús comú entre especialistes en PLN. És la unitat textual mínima processada que no sempre correspon a un terme, tal com es pot veure en la taula 4 (vegeu el *notebook* PLA-1 2.1)

Taula 4. *Tokens* d'un titular

Titular: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".	
Tokens	['The', '``', 'Daily', 'Show', "''", 'host', 'said', 'the', 'attacks', 'of', 'the', 'president', 'on', 'fellow', 'Republicans', 'at', 'a', 'meeting', 'reminded', 'him', 'of', '``', 'a', 'drunk', 'uncle', 'calling', 'everyone', 'out', 'at', 'a', 'wedding', '.', "''"]

Tal com es veu, el *tokenizer* ha distingit com a *tokens* símbols que no són termes: punts, cometes, etc. Per tant, aquests *tokens*, que comparteixen el fet que no tenen un caràcter alfanumèric, haurien de ser filtrats. Llenguatges de programació com Java, Perl o Python, i fins i tot les ordres del sistema operatiu Unix, tenen els anomenats paquets *regex* (*regular expressions* 'expressions regulars'), amb els quals es poden distingir els *tokens* que no contenen caràcters alfanumèrics. En el *notebook* PLA-1, 2.2 es pot veure com es filtren aquests *tokens* en Python.

2.2.2. Etiquetador de categoria gramatical

Beth, que també és traductora i té un coneixement lingüístic més teòric, sempre diu que la teoria aporta una visió més abstracta del problema, no deixa que un s'empantanegui en les excepcions i en els detalls i ajuda a trobar una solució més general i efectiva.

Ella suggereix un criteri per a filtrar els *tokens* que no són paraules, gràcies al qual no cal entretenir-se trobant solucions *ad hoc* per a cada excepció. El criteri és el següent: **els *tokens* que no tinguin una categoria gramatical clàssica seran filtrats**.

Quan Anne i Henry li pregunten què vol dir amb *categoria gramatical clàssica*, ella respon: «Doncs el **nom**, el **verb**, l'**adjectiu**, l'**adverbi**, el **pronom**, el **determinant**, la **preposició** i la **conjunció**».

Per a posar en pràctica aquesta idea, utilitzen el que es coneix com un *etiquetador de categoria gramatical*. Aquest etiquetador assigna a cada *token* una etiqueta que en descriu la categoria gramatical.

Els etiquetadors de categoria gramatical també es coneixen com a *PoS taggers*, i el procés d'assignació d'una etiqueta a cada *token* es coneix pel seu terme en anglès: *PoS tagging*. PoS és l'acrònim de *part of speech*, que és equivalent a 'categoria gramatical' en català.

Les etiquetes segueixen uns estàndards. L'estàndard més utilitzat és el conegut com a *Penn Treebank*, de la Universitat de Pennsilvània. En el *notebook* PLA-1, 2.3 es mostra com es fa el *PoS tagging* dels *tokens* d'un titular, i també expliquem les etiquetes estàndard que s'han assignat a aquests *tokens*. El resultat és el que apareix en la taula 5.

Taula 5. *Tokens* d'un titular etiquetats amb un *PoS tagger*

Titular: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".	
PoS tagging	[('the', 'DT'), ('``', '``'), ('daily', 'JJ'), ('show', 'NN'), ("'", "'"), ('host', 'NN'), ('said', 'VBD'), ('the', 'DT'), ('attacks', 'NNS'), ('of', 'IN'), ('the', 'DT'), ('president', 'NN'), ('on', 'IN'), ('fellow', 'JJ'), ('republicans', 'NNS'), ('at', 'IN'), ('a', 'DT'), ('meeting', 'NN'), ('reminded', 'VBD'), ('him', 'PRP'), ('of', 'IN'), ('``', '``'), ('a', 'DT'), ('drunk', 'JJ'), ('uncle', 'NN'), ('calling', 'VBG'), ('everyone', 'NN'), ('out', 'RB'), ('at', 'IN'), ('a', 'DT'), ('wedding', 'NN'), ('.', '.'), ("'", "')]

Seguint el criteri de Beth, les paraules serien els *tokens* que tenen etiquetes que comencen per *N* (nom), per *DT* (determinant) i per *V* (verb). També les preposicions (amb etiqueta «IN»), pronoms (amb etiqueta «PRP») i adverbis (amb etiqueta «RB»).

Hi ha l'etiqueta de nom propi (NPP) i hi ha etiquetadors que són sensibles al fet que el *token* tingui una majúscula inicial i etiqueten un determinant, o qualsevol *token* que no és un nom, com un nom propi. Per a evitar confusions, Beth, Anne i Henry decideixen posar els titulars en minúscula.

2.2.3. Cercador d'n-grames

Anne recorda als seus companys que queden per detectar automàticament els termes multiparaula. Anne sap que per a detectar els termes multiparaula primer s'han de trobar els n-grames del text i, d'aquests n-grames, escollir els que són més susceptibles de ser termes multiparaula. Però què és un n-grama?

Un **n-grama** és una seqüència de *tokens* consecutius que té un ordre de complexitat *n*. L'ordre de complexitat correspon al nombre de *tokens* consecutius de l'n-grama. Els n-grames d'ordre 1 es coneixen com a **unigrames**; els d'ordre 2 es coneixen com a **bigrames**, i els d'ordre 3 es coneixen com a **trigrames**.

Anne proposa assumir que els termes multiparaula més llargs en anglès són trigrames. Per això utilitzen un cercador dels bigrames i trigrames que hi ha en els titulars del *New York Times*. En el *notebook* PLA-1. 2.4 s'ensenya com es busquen els bigrames i trigrames d'un text. Els bigrames i trigrames oposats es mostren en la taula 6.

Taula 6. Bigrames i trigrames d'un titular

Titular: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".	
bigrames	[('the', '``'), ('``', 'daily'), ('daily', 'show'), ('show', """), ("""", 'host'), ('host', 'said'), ('said', 'the'), ('the', 'attacks'), ('attacks', 'of'), ('of', 'the'), ('the', 'president'), ('president', 'on'), ('on', 'fellow'), ('fellow', 'republicans'), ('republicans', 'at'), ('at', 'a'), ('a', 'meeting'), ('meeting', 'reminded'), ('reminded', 'him'), ('him', 'of'), ('of', '``'), ('``', 'a'), ('a', 'drunk'), ('drunk', 'uncle'), ('uncle', 'calling'), ('calling', 'everyone'), ('everyone', 'out'), ('out', 'at'), ('at', 'a'), ('a', 'wedding'), ('wedding', '.'), ('.', """)]
trigrames	[('the', '``', 'daily'), ('``', 'daily', 'show'), ('daily', 'show', """), ('show', """, 'host'), ("""", 'host', 'said'), ('host', 'said', 'the'), ('said', 'the', 'attacks'), ('the', 'attacks', 'of'), ('attacks', 'of', 'the'), ('of', 'the', 'president'), ('the', 'president', 'on'), ('president', 'on', 'fellow'), ('on', 'fellow', 'republicans'), ('fellow', 'republicans', 'at'), ('republicans', 'at', 'a'), ('at', 'a', 'meeting'), ('a', 'meeting', 'reminded'), ('meeting', 'reminded', 'him'), ('reminded', 'him', 'of'), ('him', 'of', '``'), ('of', '``', 'a'), ('``', 'a', 'drunk'), ('a', 'drunk', 'uncle'), ('drunk', 'uncle', 'calling'), ('uncle', 'calling', 'everyone'), ('calling', 'everyone', 'out'), ('everyone', 'out', 'at'), ('out', 'at', 'a'), ('at', 'a', 'wedding'), ('a', 'wedding', '.'), ('wedding', '.', """)]

2.2.4. Filtradors d'n-grames

Henry diu que quan Peter, el coordinador, vegi la taula d'n-grames s'espantarà molt. Tem que, quan vegi que per a un sol titular és necessari processar tants n-grames amb paraules combinades amb signes de puntuació, i amb combinacions tan poc informatives com *'out at a'*, *'republicans at'*, etc., dirà que la via que han volgut prendre és inaplicable.

Anne pensa que les combinacions amb signes de puntuació es podrien evitar si aquests signes s'agafessin del titular abans de fer la cerca d'n-grames. Beth diu que això no és correcte perquè precisament els signes de puntuació separen les paraules i les frases. L'eliminació dels signes de puntuació significaria prendre com a n-grames combinacions de paraules que eren originalment en frases diferents.

1) Filtratge amb diccionaris i altres bases de dades lèxiques

Beth planteja un filtratge d'n-grames que consisteix a prendre els n-grames d'ordre superior a 1 que apareixen com a entrades en diccionaris electrònics com Wiktionary, fonts enciclopèdiques com Wikipedia o ontologies com Wordnet. Parlarem més extensament sobre Wordnet en altres apartats, però de moment direm que és una base de dades lèxica de la llengua anglesa, desenvolupada per la Universitat de Princeton i organitzada com una ontologia. Més endavant explicarem les característiques d'una ontologia i el seu potencial.

Adaptacions de Wordnet

Actualment hi ha bases de dades lèxiques que són adaptacions del Wordnet a altres llengües. Hi ha Wordnets per a castellà, català, gallec, basc, entre altres llengües europees, finançats amb el paraigua del projecte europeu *EuroWordnet* (LI-2 4003), en el qual van participar la UPC, la UB, la Universitat del País Basc, la UNED i la Universitat de Vigo.

Anne posa algunes objeccions al plantejament de Beth. La principal objecció és que en els titulars apareixen termes multiparaula que no són recollits en cap recurs lèxic que hem esmentat, sia perquè són d'un domini molt específic o perquè són expressions que tenen un ús intensiu durant un període curt de temps i després es deixen d'usar.

Nota

Si els titulars estiguessin escrits en una llengua minoritària, l'objecció principal seria la falta de recursos lèxics en aquesta llengua per a ser consultats per una eina de processament de llenguatge natural.

2) Filtratge amb criteris tipogràfics

Les cometes marquen l'inici i el final d'un terme multiparaula, com per exemple “Daily Show”. Anne planteja buscar quadrigrames (n-grames d'ordre 4) que tinguin les cometes com a *tokens* inicial i final (p. ex., ('”', 'daily', 'show', '”')). La col·locació estaria formada pels dos *tokens* que ocupen les posicions intermèdies (‘daily show’).

Aquest filtratge té les seves limitacions, ja que no tots els autors dels titulars marquen explícitament les col·locacions amb unes cometes.

3) Filtratge amb mètodes estadístics

Henry suggereix trobar els n-grames que corresponen a termes multiparaula a partir de mètodes estadístics identificant els n-grames, els components dels quals estan relacionats tan íntimament que són considerats com a col·locacions. Segons ell, un n-grama com *'host said that'* no és un terme multiparaula que cal considerar, perquè aquesta combinació és producte de la coocurrència d'aquestes paraules en un titular determinat. És diferent el cas, per exemple, de *tea party*, en el qual les dues paraules coocorren en qualsevol text que parli d'aquest grup d'influència favorable al partit republicà.

Henry coneix càlculs estadístics com el *t-test*, el *chi-square*, el *pointwise mutual information* (PMI) o la *log-likehood* per a trobar col·locacions. Aquestes mètriques capturen la discrepància entre les probabilitats que les paraules coocorrin en qualsevol text i les probabilitats que apareguin per separat assumint que l'ocurrència d'una paraula no afecta la probabilitat d'ocurrència d'una altra paraula (són independents).

Mètriques *t-test, chi-square, PMI* i *log-likehood*

Aquestes mètriques s'expliquen en el capítol 5 del llibre de Manning i Schütze (2000), *Foundations of statistical natural language processing*. També s'expliquen en els apartats 3.1 i 6.7 del llibre de Jurafsky i Martin (2018), *Speech and language processing*.

Per tant, una col·locació es pot considerar com una seqüència de paraules que apareixen juntes molt sovint, la qual cosa és inusual. En el *notebook* PLA-1 2.5 es pot veure com es busquen els bigrames i trigrames, que són col·locacions que utilitzen la mètrica PMI. El resultat es mostra en la taula 7.

Col·locació

Segons Natural Language Processing with Python, «a collocation is a sequence of words that occur together unusually often».

Taula 7. Filtració dels n-grames d'un un titular calculant la mètrica PMI

Text: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".

	N-grames que superen el filtratge
bigrames	'calling_everyone', 'daily_show', 'drunk_uncle', 'everyone_out', 'fellow_republicans', 'host_said', 'meeting_reminded', 'on_fellow', 'president_on', 'reminded_him'
trigrames	'calling_everyone_out', 'drunk_uncle_calling', 'meeting_reminded_him', 'on_fellow_republicans', 'president_on_fellow', 'uncle_calling_everyone', 'everyone_out_at', 'fellow_republicans_at', 'reminded_him_of', 'a_drunk_uncle'

El fet de tenir un text tan curt és la causa que el càlcul estadístic del PMI no sigui prou significatiu per a evitar falsos positius (p. ex., *everyone_out*). D'altra banda, hi ha combinacions com *president_on* o *meeting reminded* que no s'haurien de considerar d'entrada termes multiparaula.

Ara bé, quan el text és gran, les coocurrències d'*everyone_out* i de *meeting reminded* perden excepcionalitat, mentre que altres coocurrències, com *fellow republicans* o *daily show*, la mantenen i són mereixedores de ser considerades col·locacions.

4) Filtratge amb *stop words*

N-grames com *president on* o *a drunk uncle* no són termes, perquè el *token* inicial o el *token* final són *stop words*. Què és un *stop word*?

> Un ***stop word*** és una paraula que té com a funció cohesionar un text però no aporta res al significat del text ni, evidentment, al significat de l'n-grama. Són *stop words* els articles, les preposicions, les conjuncions, etc.

Per tant, per a buscar termes multiparaula s'haurien de filtrar els n-grames que tenen un *stop word* a l'inici o al final. En el *notebook* PLA-1 2.6 es pot veure com es filtren els n-grames amb la llista de *stop words* de l'NLTK, que és una llibreria de Python especialitzada en processament del llenguatge natural. El resultat es mostra en la taula 8.

Taula 8. Filtratge d'n-grames amb *stop words* al principi o al final

Text: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding."

	N-grames que superen el filtratge
bigrames	'calling_everyone', 'daily_show', 'drunk_uncle', 'fellow_republicans', 'host_said', 'meeting_reminded'
trigrames	'drunk_uncle_calling', 'president_on_fellow', 'uncle_calling_everyone'

A la xarxa és fàcil trobar llistes de *stop words*, sobretot de l'anglès. Beth, però, creu que la seva utilitat depèn del criteri de les persones que les han elaborades. Per exemple, pensa que *everyone* hauria de ser en aquesta llista, i també recorda haver-hi vist paraules que no li interessava filtrar.

5) Filtratge segons la PoS dels constituents de l'n-grama

Beth creu que hi ha una alternativa al filtratge amb *stop words* que és més eficaç i que consisteix a aplicar un criteri gramatical que és el següent: es filtren els n-grames la paraula inicial o final dels quals és un determinant, pronom, verb modal, preposició, adverbi o conjunció. També considera que els verbs en gerundi[1], i flexionats[2] si estan en passat, no poden aparèixer ni al principi ni al final. En el *notebook* PLA-1 2.7 s'ensenya com s'obtenen les col·locacions segons aquest criteri. El resultat es mostra en la taula següent.

(1)Amb etiqueta VBG.

(2)Per exemple, amb etiqueta VBD.

Taula 9. Filtratge d'n-grames d'un text segons la PoS dels seus constituents

Text: The "Daily Show" host said the attacks of the president on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".

	N-grames que superen el filtratge
bigrames	'daily_show', 'drunk_uncle', 'fellow_republicans',
trigrames	'president_on_fellow', 'uncle_calling_everyone'

Veiem que el criteri de Beth no ha evitat que aparegui *everyone*. La raó és la discrepància entre el criteri de Beth i el del *PoS tagger*. Beth classificaria *everyone* com un pronom, mentre que el *PoS tagger* ho classifica com un nom. En l'apartat següent tornarem sobre la qüestió de les discrepàncies en l'etiquetatge.

2.2.5. Intèrprets i cercadors de patrons sintàctics

Beth proposa també aplicar una assumpció basada en la seva experiència com a estudiant de la gramàtica de les llengües. L'assumpció és que els termes en anglès multiparaula rellevants per al projecte seran combinacions Nom + Nom. Per aquesta raó, proposa fer una anàlisi sintàctica del titular amb un intèrpret (conegut també com a *parser*) per a identificar les combinacions de termes que tenen el patró Nom + Nom. En el *notebook* PLA-1 2.8 podem veure com s'identifiquen termes que tenen el patró sintàctic Nom i Nom + Nom. Aquests termes són els següents:

Taula 10. Termes multiparaula del text que tenen el patró Nom i Nom + Nom

Text: The "Daily Show" host said the attacks of president Trump on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".
['show', 'host', 'attacks', 'president', 'trump', 'republicans', 'meeting', 'uncle', 'everyone', 'wedding', 'president_trump']

L'intèrpret fa l'etiquetatge de categoria gramatical amb el *PoS tagger* que classifica *everyone* com un nom. Ara bé, hi ha intèrprets el *PoS tagger* dels quals pren un criteri diferent i l'etiqueta com a pronom, amb la qual cosa *everyone* no sortiria en la llista de termes.

2.2.6. Programari disponible

Hi ha una varietat de programari de codi obert amb el qual es poden llistar els *tokens*, fer l'etiquetatge de categories gramaticals, buscar els n-grames i les col·locacions i fer una anàlisi sintàctica d'un text. Tots són igualment recomanables. L'analista pot utilitzar el programari en el llenguatge de programació amb el qual se senti més còmode o li sigui més fàcil d'integrar en l'entorn d'anàlisi. A continuació presentem una taula amb alguns dels més coneguts i utilitzats.

Taula 11. Llibreries de processament del llenguatge natural de codi obert

Llibreria	Desenvolupador	Llenguatge
NLTK	Team NLTK	Python
Gensim	RaRe Technologies	Python
Spacy	Matthew Honnibal	Python i Cython
Pattern	Clips (University of Antwerp)	Python
OpenNLP	Apache Software Foundation	Java
GATE	Inicialment desenvolupat a la Universitat de Sheffield	Java
CoreNLP	Universitat de Stanford	Java
Tokenizers	Lincoln Mullen, Os Keyes, Dmitriy Selivanov, Jeffrey Arnold, Kenneth Benoit	R
MALLET	Andrew McCallum	Java

2.3. Resultats

Henry, Beth, Anne i Joseph es reuneixen amb Peter, el coordinador, per mostrar i discutir els resultats que han obtingut.

Joseph havia de veure si hi havia una relació entre la longitud dels titulars i el nombre de comentaris. La seva anàlisi no requeria un preprocessament dels titulars, ja que es limitava a tallar-los (*split*) allà on hi havia un espai en blanc i calculava la longitud de la llista d'elements separats.

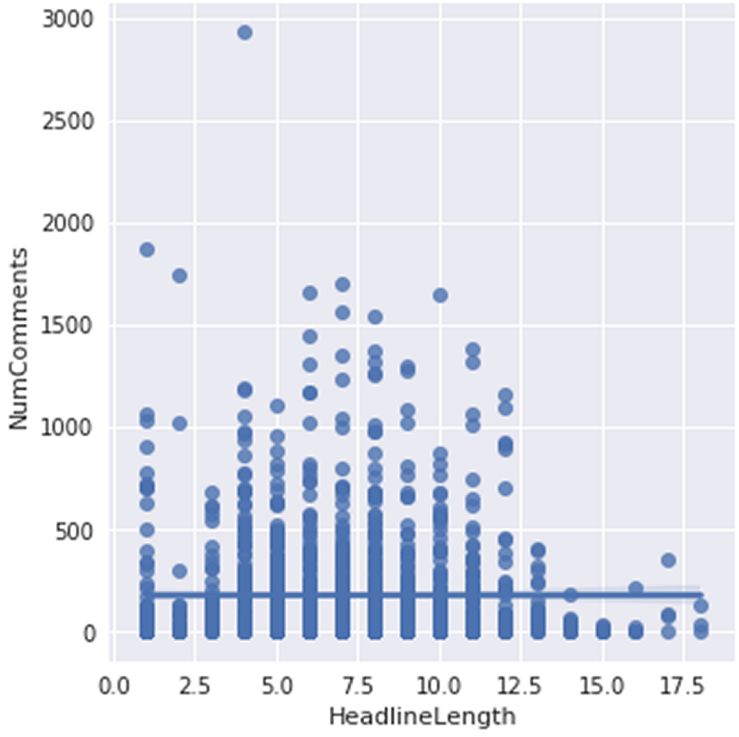
Els resultats obtinguts per Joseph mostren que la longitud dels titulars no és una dada rellevant. El grau de correlació entre la longitud i el nombre de comentaris és molt baix (figura 1) i la línia de regressió és pràcticament plana (figura 2).

Figura 1. Encapçalament del *dataframe* que relaciona la longitud dels titulars (*HeadlineLength*) amb el nombre de comentaris (*NumComments*) i el grau de correlació.

	articleID	HeadlineLength	NumComments
0	5a974697410cf7000162e8a4	4	37
1	5a974be7410cf7000162e8af	7	22
2	5a9752a2410cf7000162e8ba	7	307
3	5a975310410cf7000162e8bd	9	44
4	5a975406410cf7000162e8c3	11	365

	HeadlineLength	NumComments
HeadlineLength	1.000000	-0.000054
NumComments	-0.000054	1.000000

Figura 2. Representació gràfica de la correlació entre la longitud dels titulars (*HeadlineLength*) i el nombre de comentaris (*NumComments*)



Per la seva banda, Anne, Beth i Henry expliquen en la reunió que han treballat en el preprocessament dels titulars i en l'elaboració d'algorismes per a detectar els termes monoparaula i multiparaula. Els queda per fer l'exploració dels termes i detectar els que són més rellevants per a relacionar els titulars amb el nombre de comentaris.

Vistos els resultats obtinguts amb el no-processament dels titulars, Peter diu a tots els membres del grup que s'involucrin en l'exploració de termes i queden per a la setmana següent per discutir els resultats obtinguts.

3. Detecció de termes rellevants

En aquest apartat presentarem els aspectes que cal tenir en compte a l'hora de detectar i processar les paraules més rellevants per a l'anàlisi. En el nostre cas d'ús, les paraules més rellevants són les dels titulars que influeixen suposadament en el lector i el motiven a escriure comentaris. Com en l'apartat anterior, exposarem aquests aspectes i també els recursos disponibles explicant com els membres de S&S es van enfrontant als problemes més comuns.

3.1. Llei de Zipf

Anne, Beth, Joseph i Henry són conscients que s'han d'enfrontar a l'anomenada **llei de Zipf**, una llei que contradiu el mite que les dades rellevants estan entre les dades més freqüents. Aquest mite pot ser cert per a molts tipus de dades, però quan les dades són lingüístiques no funciona.

El filòleg i lingüista George Kingsley Zipf va presentar, a mitjan anys quaranta del segle passat, la llei que porta el seu nom i que pot resumir-se de la manera següent.

Llei de Zipf

Si llistem les paraules segons l'ordre de freqüència, això és, pel rang, la segona paraula de la llista apareixerà aproximadament la meitat de vegades que la primera paraula, la tercera paraula apareixerà amb una freqüència aproximada d'un terç de vegades, i així successivament.

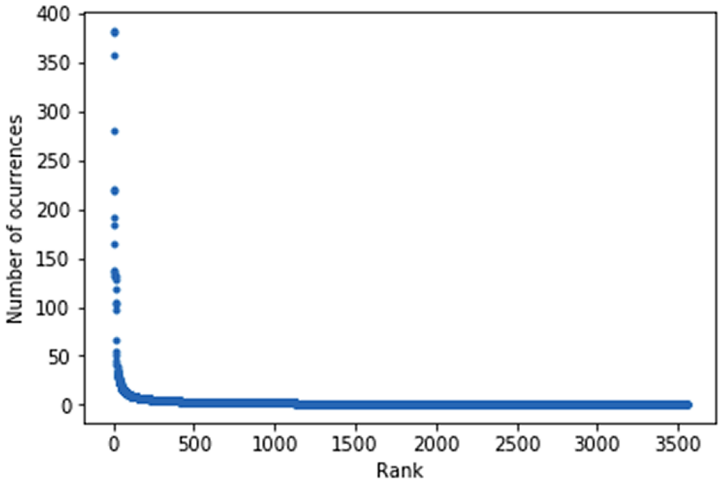
Segons aquesta llei, les paraules que ocupen les primeres posicions concentren el major percentatge de les paraules d'un text. El que té de rellevant és que entre aquestes paraules hi ha els articles, les preposicions o les conjuncions, paraules que, malgrat repetir-se molt en el text, no aporten gairebé gens de contingut.

Zipf va formular la seva llei per demostrar la tendència que tenen els humans a fer les coses amb el mínim esforç. Si el receptor és capaç d'entendre el missatge quan l'emissor utilitza les paraules més importants N vegades, per què repetir-les més vegades? Les paraules que es repeteixen molt són aquelles la funció de les quals és articular el missatge de manera recognoscible i comprensible per al receptor. Per exemple, que el receptor sàpiga en quina llengua li estan parlant, aquesta és la funció principal dels articles, preposicions, etc.

La llei es compleix amb un llibre, un article, una notícia, etc. I els titulars del *NewYorkTimes* no en són una excepció. La gràfica[3] següent mostra la relació entre la freqüència i el rang dels termes monoparaula dels titulars del *New York Times*. El rang és la posició que un terme ocupa en la llista de termes ordenats per freqüència.

(3)En el *notebook* PLA-1. 3.1 hi ha l'*script* que genera la gràfica.

Figura 3. Relació entre la freqüència i el rang dels termes monoparaula en els titulars del *New York Times*



Les paraules que tenen més informació sempre són en la part més baixa de la corba. Fins i tot poden ser *hapax legomenon*, que significa 'dit una sola vegada' en grec, ja que, per què haig de repetir una vegada i una altra una paraula si, dita una sola vegada, ja queda clar quin és el missatge que vull transmetre?

3.2. Filtratge de *stop words*

Les paraules que concentren el major percentatge de freqüència d'aparició, però que no aporten gens de contingut, són les *stop words* (vegeu l'apartat 2.2.4.4). Per tant, aquestes paraules s'haurien de filtrar abans de procedir a l'exploració de les dades interessants.

3.3. Normalització del text

Una tasca important del procés d'exploració de les dades és la **normalització** del text. Els elements d'un text són termes com *llibre* i *llibres*, que són variants d'una forma que aglutina les referències al concepte de llibre. És important, per tant, que les dades que cal explorar siguin les formes aglutinadores de les variants.

3.3.1. Lemes i *stems*

La forma que aglutina les variants sol ser el lema.

El **lema** és la forma de les entrades d'un diccionari. Quan es tracta d'un nom, en llengües com l'anglès, el castellà, el català, etc. el lema coincideix amb la forma singular. Quan es tracta d'un verb, el lema coincideix amb l'infinitiu.

Col·loquialment, quan algú diu que la paraula *P* significa *X*, identifica la paraula amb el seu lema.

A partir d'ara, i per a facilitar la comprensió, quan diguem «la paraula *P* apareix *N* vegades», ho farem en el sentit col·loquial. Així, quan diguem que la paraula *kill* apareix dues-centes vegades, és perquè hem sumat les vegades que apareixen les variants del lema *kill* (*killed, kill, kills, killing*, etc.)

Una altra forma normalitzada és l'**arrel** d'un terme (*stem* en anglès).

L'**arrel** d'un substantiu, i també l'arrel d'un verb, és el que queda després d'haver-ne tret els morfemes variables. El procés de treure els morfemes variables és conegut com a *stemming*.

Hi ha diferents algorismes de *stemming*, per la qual cosa els resultats poden variar segons l'algorisme triat.

A continuació mostrem uns exemples de lematització i *stemming* (vegeu el *notebook* PLA-1 3.2). L'algorisme de *stemming* utilitzat és el *Porter stemmer*.

Taula 12. Exemples de lematització i *stemming*

Formes	Lema	Stem
feet	*foot*	*feet*
elephants	*elephant*	*eleph*
communities	*community*	*commun*

3.3.2. Sinonímia

Anne creu que la normalització de les dades pot anar més enllà agrupant termes pel seu significat. El criteri més evident d'agrupació de termes segons el seu significat és la relació de sinonímia. És a dir, Anne pensa a processar expressions variants del mateix significat i posa com a exemple explicar formes variants del concepte de matar. D'aquesta manera, s'explicarien termes monoparaula i multiparaula com *kill* i *put_to_death* com a variants d'aquest concepte, igual que s'explicarien *President Trump*, *President Donald Trump* o *Donald Trump* com a referències al mateix personatge.

Anne sap molt bé la importància que té la sinonímia, ja que recorda la seva anàlisi amb mètodes de PLN de les ofertes laborals en intel·ligència de negoci. Anne no es podia imaginar que la sinonímia li comportaria un veritable maldecap. Unes ofertes de treball posaven *business intelligence*, altres *BI* i altres *Business Analytics*; també hi havia qui posava *DW* en comptes de *Data Warehouse*, etc. Es va horroritzar quan Peter li va dir que s'estava parlant de fer la mateixa anàlisi per a una empresa espanyola amb ofertes publicades a Espanya i també amb ofertes publicades en països de Sud-amèrica, la qual cosa comportava unificar la referència als mateixos conceptes en totes les variants de l'espanyol. Finalment el projecte no es va dur a terme.

3.3.3. Hiperonímia

Beth, sempre disposada a aplicar els seus coneixements de teoria lingüística, proposa explorar un altre criteri de normalització basat en el significat. Aquest criteri és la relació d'hiperonímia.

En una ontologia, la **relació d'hiperonímia** és la que s'estableix entre les instàncies d'una classe i la classe. Aquesta relació és del tipus *A* és un *B*.

Per exemple, les paraules *pistola*, *rifle* i *escopeta* comparteixen el mateix hiperònim, ***arma***, amb la qual cosa aquestes paraules es podrien agrupar en un nivell més abstracte que el de la sinonímia. Les paraules que comparteixen un hiperònim *H* es coneixen com a **hipònims** de *H*.

Beth creu que les dades rellevants per al projecte no són tant els termes com els hiperònims. Considera que els hiperònims són com els temes dels quals parla un titular, i que caldria orientar el projecte a poder fer afirmacions com aquesta: «Quan es parla d'armes o esports hi ha més comentaris».

A continuació presentem un exemple de normalització de termes que són *N* i *N* + *N*. Els termes *attacks* i *fire*[4] es normalitzen en un sol terme per compartir el mateix hiperònim.

(4)*Fire* té el sentit de 'crítica' en aquest exemple, un sentit que no és el que ens ve a la ment en primer lloc quan la paraula està sense context.

Taula 13. Normalització de termes per hiperonímia

Text: The "Daily Show" host said the attacks and fire of president Trump on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".	
	Termes
Sense normalització per hiperonímia	['show', 'host', 'attacks', 'fire', 'president', 'trump', 'republicans', 'meeting', 'uncle', 'everyone', 'wedding', 'president_trump']
Normalització per hiperonímia (hiperònim comú: *criticism*)	['show', 'host', '**criticism**', 'president', 'trump', 'republicans', 'meeting', 'uncle', 'everyone', 'wedding', 'president_trump']

Vegeu també

Sobre el sentit de les paraules, vegeu l'apartat 3.3.4.

3.3.4. El problema de l'ambigüitat

La normalització automàtica del text pot produir resultats no desitjats, en gran part a causa de l'ambigüitat en la forma de les paraules.

Dues paraules poden tenir la mateixa forma però la categoria gramatical i el significat diferents.

> Per exemple, en la frase *The inhabitants house the inmigrants in their own house* la forma *house* té dues PoS diferents. El primer *house* és un verb i el segon *house* és un substantiu. El verb *house* significa 'allotjar' i el substantiu significa 'casa'. Per tant, el lema *house* no hauria d'aglutinar aquestes dues formes.

D'altra banda, dues formes idèntiques amb la mateixa PoS poden tenir significats diferents.

> Per exemple, tenim *run*, que és la forma de dos verbs diferents. El verb *run* pot significar 'córrer' i també pot significar 'executar', com en *this software can run several tasks at the same time*.

La distinció de les PoS i dels sentits de les paraules que tenen la mateixa forma es coneix com a **desambiguació** i, en l'argot de la lingüística computacional, *word sense disambiguation* o WSD.

La normalització depèn, per tant, de bones eines de desambiguació, que és una de les tasques més difícils en el processament del llenguatge natural.

3.3.5. Recursos

La lematització es pot fer amb el paquet NLTK, que busca en la base de dades lèxica Wordnet el lema d'un terme (vegeu PLA-1. 3.2). També es pot fer amb **la llibreria Wordnet** en R. Si el terme té la mateixa forma en més d'una categoria gramatical, cal especificar al lematitzador la categoria gramatical que el *PoS tagger* ha assignat al terme.

Wordnet, com a base de dades lèxica organitzada com una ontologia, és a més una eina bàsica per a desenvolupar la normalització dels termes segons les relacions de sinonímia i hiperonímia.

Quant a l'*stemming*, cal destacar l'stemmer **Snowball**. Snowball fa *stemming* de paraules en anglès, català, espanyol, francès, italià, a més d'altres llengües romàniques i germàniques, i també fa *stemming* de l'àrab. Està integrat en la llibreria NLTK, encara que també està en Java, integrat en el cercador Lucene.

La desambiguació o WSD és una tasca que han de fer els intèrprets, ja que una representació sintàctica correcta depèn de la qualitat de desambiguació de la PoS i el sentit de les paraules.

Actualment, un dels intèrprets de codi obert més utilitzat és **FreeLIng**. Al seu torn, la majoria d'eines presentades en 2.2.6 tenen un intèrpret.

FreeLing

FreeLing, desenvolupat per Lluís Padró a la UPC, analitza, a més d'anglès i espanyol, textos en català, basc i gallec, entre altres llengües. Vegeu **http://nlp.lsi.upc.edu/freeling** i **X. Carreras; I. Chao; L. Padró** i **M. Padró** (2004). «FreeLing: An Open-Source Suite of Language Analyzers». *Proceedings of the 4th International Conference on Language Resources and Evaluation*.

Amb la llibreries en Python NLTK i SpaCy, s'analitzen textos en anglès, encara que SpaCy pot aprendre a analitzar en una altra llengua si disposa d'un model d'aquesta llengua. Pattern també analitza textos en espanyol i GATE té el seu intèrpret per a l'anglès en llenguatge Java.

3.3.6. Resultats

Arriba el dia en què han de mostrar a Peter els resultats obtinguts. Solament han tingut temps de normalitzar el text via lematització. També han unificat termes com *Mr. Trump* o *Donald Trump* amb el terme aglutinador *Trump*. Han preferit la lematització a l'*stemming* perquè els lemes són recognoscibles més fàcilment a l'hora de verificar el procés de normalització del text que les arrels de les paraules, que sovint poden ser confuses. La normalització per mitjà de relacions semàntiques com la sinonímia, la hiperonímia, etc., proposada per Beth, es presenta com una possibilitat que cal considerar.

Bibliografia

Sobre les relacions de sinonímia, hiperonímia i altres, vegeu el capítol 6 del llibre de **Jurafsky** i **Martin** (2018) *Speech and Language Processing*.

Anne, Beth i Henry presenten un *tag cloud* (figura 4) de la freqüència dels termes monoparaula i multiparaula lematitzats —en el *notebook* PLA-1 3.3 es mostra com es fa. A Peter li crida l'atenció que *Trump* i *president* estiguin entre els termes més freqüents. També considera que hi ha molts termes que per si sols no aporten res (p. ex., *say, go, new*), perquè poden aparèixer en tots els titulars, motivadors de comentaris o no, i hi ha altres termes que possiblement són subalterns de termes que són més rellevants i haurien d'estar més destacats. Per exemple, li queda el dubte de si *president* és un terme vinculat a *Trump* o bé és un terme que indica que els presidents, siguin quins siguin, provoquen comentaris en els lectors.

Per tant, Peter els diu que treballin per a trobar les paraules realment rellevants i els cita per a la setmana següent.

Figura 4. Núvol d'etiquetes dels termes lematitzats en els titulars del *New York Times*

3.4. Vectorització del text

Quan es reuneix l'equip, Henry apunta que l'encàrrec de Peter comporta anar més enllà d'explicar termes. Els termes s'han de posar en el context de dos tipus de titulars. El primer tipus de titular és el que provoca molts comentaris, i el segon és el que provoca pocs comentaris. Per tant, cal trobar els termes que són rellevants en els titulars del primer tipus, i això significa que cal representar-los d'una manera computacionalment tractable. La representació dels textos com a vectors és la més utilitzada.

3.4.1. Vectoritzador

El vectoritzador, més conegut pel seu terme en anglès *vectorizer*, és l'eina que transforma cada text d'una col·lecció de documents en un vector. La vectorització es fa amb una matriu de termes i documents (*term-document matrix*) i amb el conjunt de termes diferents que apareixen en tots els documents de la col·lecció (vocabulari).

Cada columna de la matriu representa un document de la col·lecció, i cada fila és la representació d'un terme del vocabulari. Quan el vectoritzador es basa en la freqüència dels termes (*count vectorizer*), cada cel·la d'aquesta fila representa les vegades que un terme del vocabulari apareix en el document de la columna.

La taula 14 mostra el *term-document-matrix* de quatre obres de Shakespeare suposant que el vocabulari d'aquestes obres és de quatre paraules[5].

[5] L'exemple és la figura 6.2 del llibre de Jurafsky i Martin (2018).

Taula 14. Exemple de *term-document-matrix*

	As you like it	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	5	117	0	0

La vectorització de cada obra es fa recollint els valors numèrics de la columna corresponent. És a dir, *Henry V* es representaria amb el vector [15, 36, 5, 0] i *Julius Caesar* amb el vector [8, 12, 1, 0].

El *term-document-matrix* permet descriure el document com un vector, però, a més, descriure un terme del vocabulari també com un vector. El vector [5, 117, 0, 0], que representa *clown*, descriu la contribució d'aquest terme en les quatre obres.

Vegem ara com es vectoritzen quatre titulars del *New York Times* amb un *count-vectorizer*. Els titulars són els següents:

T1: Trump proclaims tariffs on steel, and a tariff on aluminium and stocks sag in reply

T2: Trump embraces a trade war, which could undermine growth

T3: Mr. Trump and destructive trade war

T4: The macroeconomics of trade war

El *term-document matrix* és el que apareix a continuació. Els termes considerats són de la forma N i $N + N$, i han estat lematitzats.

Taula 15. *Term-document-matrix* de quatre titulars del *New York Times*

	T1	T2	T3	T4
Trump	1	1	1	0
tariff	2	0	0	0
steel	1	0	0	0
stock	1	0	0	0
reply	1	0	0	0
trade	0	1	1	1
war	0	1	1	1
growth	0	1	0	0
trade_war	0	1	1	1
aluminium	1	0	0	0
macroeconomics	0	0	0	1

Els vectors que representen els titulars són els següents:

T1: [1, 2, 1, 1, 1, 0, 0, 1, 0, 1, 0]

T2: [1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0]

T3: [1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0]

T4: [0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 1]

3.4.2. Vectoritzador tf.idf

A partir d'aquest moment, els titulars de les notícies amb més comentaris es diran titulars TOP. Els titulars restants seran titulars NOTOP. Els titulars del tipus TOP són els de les notícies amb un nombre de comentaris que superen la mitjana.

Per a Henry, el vectoritzador no hauria de tenir en compte solament la presència dels termes del vocabulari en els titulars TOP; també hauria de tenir en compte la presència dels termes del vocabulari en els titulars NOTOP. Posa com a exemple un doctorand que fa la tesi comparant els trets estilístics de *Moby Dick* de Herman Melville amb els de *Sense and Sensibility* de Jane Austen. És evident que ha de buscar termes que apareixen a *Moby Dick*, com *white whale* o *wildest watery spaces*, però també ha de comprovar que aquests són termes característics d'aquesta novel·la perquè no apareixen a *Sense and Sensibility*. Peter ja apuntava una qüestió semblant quan deia que *president* i *new* no semblaven ser termes específics dels titulars TOP, perquè poden aparèixer també en els titulars NOTOP.

Henry proposa vectoritzar els titulars amb l'anomenat *vectoritzador tf.idf*. El vectoritzador tf.idf s'aplicaria a partir d'un *term-document-matrix* amb dues columnes: una per als titulars TOP i una altra per als titulars NOTOP. Les files descriurien el vocabulari de termes de tots els titulars i cada cel·la contindria un valor numèric indicatiu de la contribució d'un terme als titulars TOP i NOTOP. La contribució no és la freqüència del terme en un tipus de titular, sinó el valor d'una mètrica anomenada *tf.idf*.

La idea de la mètrica tf.idf és la següent: donat un document *d* en una col·lecció de documents, un terme *t* és característic del document *d* si *t* és freqüent en *d*, però, en canvi, no està o apareix molt poc en la resta de documents de la col·lecció. Tornant a l'anàlisi de *Moby Dick*, el terme *white whale* és representatiu de la novel·la de Melville, perquè és molt freqüent a *MobyDick*, però no ho és a *Sense and Sensibility*.

La mètrica és la combinació d'altres dues mètriques: el tf, o *term frequency*, que és la freqüència total de *t* en *d*, i l'idf, *inverse document frequency*, que és la inversa del nombre de documents de la col·lecció en els quals apareix.

Bibliografia

Sobre el pes i la rellevància dels termes segons el seu tf.idf, vegeu l'apartat 15.2.2 del llibre de Manning i Schütze (2000) i també els apartats 6.5 i 6.6 del llibre de Jurafsky i Martin (2018). En l'entrada «tf.idf» de la Wikipedia hi ha una explicació extensa de la fórmula i un exemple concret. (https://en.wikipedia.org/wiki/tf–idf)

La fórmula és la següent:

$$\text{tf.idf}(t, d, D) = \text{tf}(t, d) * \text{idf}(t, D)$$

t: terme; *d*: document; *D*: col·lecció de documents

Per a calcular el TF, sovint es normalitza la freqüència d'un terme en el document amb el nombre total de paraules del document.

L'IDF és el logaritme de la divisió entre el nombre de documents de la col·lecció i el nombre de documents de la col·lecció que tenen el terme. El resultat és un nombre del 0 a l'1. 0 significa que el terme no és gens rellevant en el document, mentre que 1 significa que la rellevància del terme en el document és màxima.

Nota

Com que els termes que apareixen en tots els documents tindrien un valor d'IDF igual a 0, perquè el logaritme d'1 és 0, per a evitar casos de multiplicació per 0 podem trobar una versió de la fórmula en la qual l'IDF es descriu com a $1 + \log(N/n)$, on N és el nombre total de documents i n el nombre de documents que contenen el terme t.

3.4.3. Qüestions relatives a la vectorització

Un vectoritzador té un component anomenat *analitzador* (*analyzer*), que s'ocupa d'establir les unitats sobre les quals es calcularà un valor numèric (freqüència o tf.idf). És important assenyalar que l'analitzador estableix les unitats d'una col·lecció de textos. Les unitats són generalment n-grames que compleixen unes condicions (per exemple, que no siguin *stop words*), encara que poden ser altres unitats definides per l'analista (per exemple, termes amb el *pattern* «N i N + N», tal com hem vist en la vectorització de titulars en 3.4.1).

La llista d'unitats establertes per l'analitzador és el vocabulari de la col·lecció de textos. Cada unitat o terme ocupa una posició en la llista. La posició o índex del terme no està preestablert perquè el vocabulari serveix per a representar els textos segons un model conegut com a *bag of words* o BOW, en el qual l'ordre en què apareixen les unitats no és important. El *bag of words* és una metàfora molt gràfica per a entendre els textos com un conjunt de termes sense un ordre establert (figura 5).

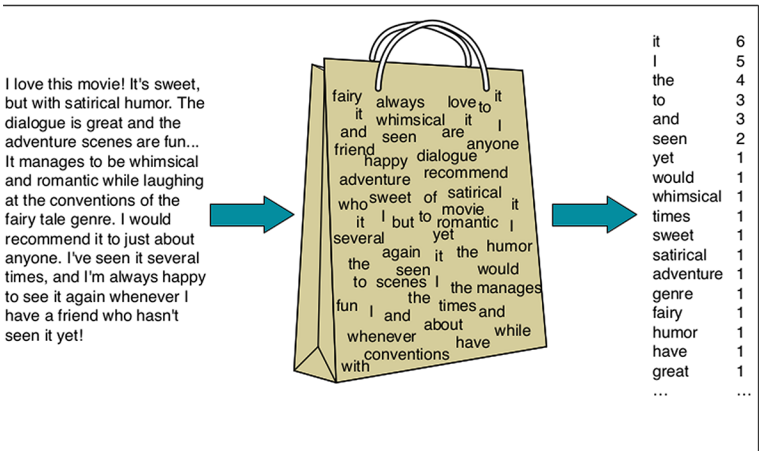
Bibliografia

Sobre la representació BOW, consulteu l'apartat 4.1 del llibre de **Jurafsky** i **Martin** (2018).

Una vegada obtingut el vocabulari, el vectoritzador construeix, per a cada text de la col·lecció, un vector amb tantes dimensions com índexs hi ha en el vocabulari. El vectoritzador calcula, per a cada terme del vocabulari, un valor numèric que el representa. Aquest valor numèric pot ser, tal com s'ha dit, la freqüència del terme en el text que es vectoritza o el seu valor de tf.idf. El valor

obtingut es posa en el vector en la mateixa posició que ocupava el terme en el vocabulari, és a dir, el seu índex. El valor numèric és la *feature* i el terme que la *feature* descriu és la *feature name*.

Figura 5. Il·lustració d'un text com un *bag of words* amb la freqüència de cada unitat, en aquest cas paraula



Font: Jurafsky i Martin (2018), pàg. 65

La taula següent il·lustra la vectorització d'una col·lecció amb dos titulars, T1 i T2, que hem vist anteriorment (c.f. 3.4.1). El vectoritzador és un tf.idf (vegeu l'exemple en el *notebook* PLA-1 3.4)

Taula 16. Vectorització d'una col·lecció de dos documents amb un *tf.idf vectorizer*

Unitats detectades per l'*analyzer*	*pattern N, N + N*		
Titulars	T1: *Trump proclaims tariffs on steel, and a tariff on aluminium and stocks sag in reply* T2: *Trump embraces a trade war, which could undermine growth*		
Vocabulari	'aluminium', 'growth', 'proclaims*', 'reply', 'steel', 'stock', 'tariff', 'trade', 'trade_war', 'trump', 'war'		
Text	**Índex**	***Feature name***	**Vector de *features***
T1	0 1 2 3 4 5 6 7 8 9 10	'aluminium', 'growth' 'proclaims' 'reply' 'steel' 'stocks' 'tariff', 'trade' 'trade_war' 'trump' 'war'	[0.32 0 0.32 0.32 0.32 0.32 0.64 0 0 0.23 0]

* Problema de WSD. L'intèrpret ha pres *proclaims* com un nom i no com un verb.

			[
T2	0	'aluminium',	0.0
	1	‘growth’	0.47
	2	'proclaims'	0.0
	3	'reply'	0.0
	4	'steel'	0.0
	5	'stocks'	0.0
	67	'tariff',	0.0
	8910	'trade'	0.47
		‘trade_war’	0.47
		'trump'	0.33
		‘war’	0.47
			]

* Problema de WSD. L'intèrpret ha pres *proclaims* com un nom i no com un verb.

És important tenir en compte que un vector que representa un document té molts zeros, perquè un gran percentatge de les paraules del vocabulari no són presents. Tenint en compte que la longitud d'un vector és el nombre de termes del vocabulari, les longituds dels vectors poden ser molt grans, sobretot quan el vocabulari és ingent, com el de la Wikipedia sencera o tots els articles del *New York Times*. Per aquesta raó, se sol posar un límit de *features* (*max_features*) per reduir el vector a les *features* que tenen un pes més gran.

3.4.4. Recursos

Generalment, les llibreries especialitzades en aprenentatge automàtic tenen vectoritzadors. En Python, una de les llibreries de referència és Sklearn.

En Sklearn, el *term-document-matrix* és diferent del mostrat en les taules 1.14 i 1.15. El corpus de documents es representa com una matriu amb una fila per a cada document i una columna per a cada n-grama, *pattern*, etc. del vocabulari de documents.

3.4.5. Resultats

En la reunió amb Peter, Henry, Anne, Beth i Joseph presenten els resultats d'haver vectoritzat un document que recull tots els titulars TOP amb un vectoritzador tf.idf. La taula següent mostra els deu primers termes dels titulars TOP ordenats pel seu valor de tf.idf. Els resultats mostren que *Trump* és el terme més significatiu i característic dels titulars TOP.

Figura 6. Deu primers termes dels titulars TOP ordenats pel seu tf.idf

	Term	TfIdf
0	trump	0.837212
1	president	0.123119
2	school	0.098496
3	tariff	0.098496
4	say	0.086184
5	trade	0.086184
6	ex	0.073872
7	facebook	0.073872
8	get	0.073872
9	new	0.073872

Peter comenta que li sorprèn que la relació entre un titular i el nombre de comentaris sigui tan clara solament amb un terme. Creu que ha d'haver-hi quelcom més, tal vegada relacionat amb la presència de paraules com *trade* o *tariff*. De moment, li és difícil saber si els lectors del *New York Times* s'animen a comentar-ho pel sol fet de veure la paraula *Trump* en el titular, o bé hi ha altres factors que les altres paraules sense context no tenen capacitat de revelar de manera clara. Peter vol que li resolguin aquest dubte per a la propera reunió.

Abans de sortir del despatx, Peter els diu que potser la clau és conèixer les connotacions que té la paraula *Trump* en els titulars del *New York Times*. Conèixer una paraula no es limita a denotar-la. Una paraula com *Trump* s'ha de conèixer en el context de l'ideari del diari. Al cap d'una estona d'estar parlant sobre com es coneix una paraula, Peter els diu: «No sé si ve al cas, però recordo que una vegada el lingüista J. R. Firth va dir que una paraula se la coneix per les seves companyies, una idea que es remunta a Ludwig Wittgenstein». Sense donar molta importància al que pensen, ja que és una cita més a la qual Peter els té acostumats, els membres de l'equip decideixen anar a dinar.

3.5. Vectorització dels termes

Quan l'equip es reuneix, Joseph els recorda el que Peter els va dir sobre el fet que una paraula se la coneix per les seves companyies. Companyia implica proximitat, per la qual cosa, si cal conèixer la paraula *Trump* pels titulars del *New York Times*, han de saber quines paraules li són properes.

La proximitat comporta situar elements a l'espai i, en realitat, tal com reconeix Henry, és el que han fet quan han vectoritzat els titulars. Han posat aquests vectors en un espai, anomenat *espai vectorial*. El mateix s'hauria de fer amb el terme *Trump* i amb els termes dels titulars de notícia que provoquen més

comentaris: caldria vectoritzar els termes i situar-los en un espai vectorial on poder veure els termes propers. La pregunta és: com es pot convertir un terme en un vector?

3.5.1. *One-hot vector*

Una manera directa de convertir un terme en un vector és convertir-lo en un *one-hot vector*.

> Un *one-hot vector* és un vector d'*N* dimensions amb una sola dimensió amb el valor d'1; la resta són 0.

Un terme *t* es pot representar com un *one-hot vector* amb tantes dimensions com termes en el vocabulari. El vector té zeros en totes les dimensions excepte en la dimensió corresponent a l'índex de *t* en el vocabulari.

Imaginem un vocabulari de deu termes i que el terme *Trump* té l'índex número tres en el vocabulari. La figura 7 mostra el *one-hot vector* que representa *Trump*.

Figura 7. *One-hot vector* que representa el terme *Trump* en un vocabulari de deu termes

0	0	1	0	0	0	0	0	0	0

Imaginem ara com serien els *one-hot vectors* dels termes dels titulars del *New York Times* i dels termes de la Wikipedia: vectors de centenars de milers, fins i tot milions, de dimensions, totes amb 0 i un sol 1.

3.5.2. *Word embeddings*

> Convertir un terme en un vector i posar-lo en un espai vectorial es diu *word embedding*. En el vector que representa un document, en l'índex corresponent a un terme s'insereix (*embed*) el vector que representa la paraula.

Quan els vectors tenen una dimensionalitat molt gran perquè el vocabulari és molt extens, processar *one-hot vectors* té un cost computacional que no correspon a les dades que es poden obtenir.

Per exemple, seria esperable que els vectors de dos termes sinònims fossin iguals o, com a mínim, molt propers. No obstant això, els *one-hot vectors* de dos termes sinònims poden ser molt diferents, perquè l'única dimensió que no és zero pot ocupar índexs dispars en els vectors que representen els dos termes.

3.5.3. *Dense vectors*

L'objectiu és representar els termes amb vectors de dimensionalitat més reduïda que, a més, serveixin per a capturar les relacions entre els termes del vocabulari.

Els vectors de dimensionalitat reduïda es diuen *dense vectors* i es construeixen segons la relació de coocurrència del terme amb els termes del vocabulari en un context.

Un mètode clàssic de creació de *dense vectors* és l'SVD (*singular value decomposition*). Aquest mètode rota els eixos de l'espai on són els vectors. El resultat de la rotació és un nou espai, de manera que la primera dimensió del nou espai mostra la variació major. La generació d'espais nous i rotacions posteriors fa que, amb un nombre més reduït de dimensions, s'obtingui una representació significativa de les dades malgrat perdre informació. Una de les aplicacions més conegudes de l'SVD és el *latent semantic analysis* (LSA). La freqüència de coaparició d'un terme amb els altres termes del vocabulari en un context és el criteri de generació de nous espais amb major variabilitat.

Bibliografia

En el capítol 6 del llibre de **Jurafsik** i **Martin** es pot trobar més informació sobre *word embeddings* i *dense vectors*. La figura 16.1 del mateix llibre, en l'edició del 2015, il·lustra molt bé la creació de *dense vectors*.

L'apartat 6.8 del llibre de **Jurafsik** i **Martin** (2018) explica el mètode *latent semantic analysis* (LSA) de manera més detallada.

3.5.4. Word2Vec

Word2Vec és un mètode que recentment ha tingut un gran impacte. D'una banda, redueix la dimensió dels vectors que representen els termes i, d'altra banda, captura la relació semàntica entre els termes del vocabulari. A més, és un mètode molt suggeridor, perquè es basa en un procés cognitiu que s'ha modelat mitjançant una xarxa neuronal i és representant d'un corrent de l'aprenentatge automàtic, anomenat aprenentatge profund, amb projecció en la intel·ligència artificial.

Enllaç d'interès

Per a veure com s'implementa el Word2Vec amb una xarxa neuronal: https://towardsdatascience.com/learn-word2vec-by-implementing-it-in-tensorflow-45641adaf2ac.

Si llegim *United States of*, completem inconscientment la frase amb la paraula *America*. L'eina que aplica el Word2Vec també. És a dir, després d'haver estat entrenada amb moltíssims documents, l'eina, com nosaltres, aprèn a generar uns *dense vectors* basats en la probabilitat que, donada la presència de *United States of*, aparegui el terme del vocabulari *America*. Així aprèn a modelar els textos del document. Concretament, aquest mètode de modelatge es diu *continuous bag of words* (CBOW). També aplica el mètode conegut com a *skip-gram*, que calcula la probabilitat que, donada una paraula, aparegui una combinació determinada de termes del vocabulari. Per exemple, donada la paraula *Suprem*, calcula la probabilitat que la precedeixi *el Tribunal*.

El que ha provocat més impacte ha estat el descobriment de les virtuts que tenen els *dense vectors* calculats amb aquests dos mètodes. Situats en un espai vectorial, els vectors representen molt bé el significat dels termes i les relacions semàntiques entre ells. Per exemple, els vectors dels termes sinònims (gairebé)

se superposen i els termes semànticament allunyats també estan superposats a l'espai vectorial. Per la seva banda, aplicant operacions de suma o diferència entre vectors s'obtenen vectors semànticament propers a altres termes, amb la qual cosa es poden fer inferències que relacionen *king* i *queen*, o *Roma* amb *París* i *França* amb *Itàlia*. Aquesta capacitat de fer inferències amb simples operacions sobre vectors fa que Word2Vec s'hagi convertit en un mètode estàndard per a descobrir relacions que no són evidents.

Aquestes operacions representen termes en el mètode Word2Vec:

VECTOR(king) – **VECTOR('woman')** = vector cercano a **VECTOR('queen')**

VECTOR(París) – **VECTOR (Francia)** + **VECTOR(Italia)** = vector cercano a **VECTOR(Roma)**

Els resultats, sobretot quan el corpus és gran, confirmen el principi que un terme és semblat a un altre si els dos comparteixen contextos similars.

> En la frase «el hijacok té un gust dolç», encara que no sapiguem què és *hijacok*, intuïm que és un aliment, perquè comparteix context amb paraules com *poma* o altres termes de menjar.

S'han fet estudis per veure si en altres dominis també es compleix aquest principi, que defineix, per exemple, *context* com 'la distribució de la població en una zona geogràfica'. D'aquesta manera, es pot aplicar a diferents dominis el potencial predictiu i de descobriment de relacions no evidents que té aquest mètode (recomanadors, traducció automàtica, descobriment de casuístiques «amagades» en informes mèdics, etc.)

3.5.5. Eines per a fer *word embeddings*

Word2Vec és el nom de l'eina que Google va desenvolupar en C++ per aplicar el seu mètode. Hi ha una versió en Python que està en la llibreria **Gensim**.

Una altra aplicació del *word embeddings* és **Glove** (Global Vectors for Word Representation), de la Universitat de Stanford, també disponible en Python i en Java (JGloVe). La diferència respecte a Word2Vec és que Glove explica els contextos coocurrents (*context-counting vectors*), mentre que **Word2Vec** els prediu (*context-predicting vectors*).

La plataforma **H20** també permet executar Word2Vec en R.

Facebook també ha creat una llibreria per a fer *word embeddings*. Es diu **Fasttext** i té la particularitat que pren combinacions de caràcters (n-grames de caràcters) com una unitat mínima per a fer els *word embeddings*. També està disponible en Python.

Per a abordar reptes del processament del llenguatge natural aplicant aprenentatge profund, cal destacar la llibreria de font pública TensorFlow, distribuïda per Google en llenguatge Python. TensorFlow no solament és útil per a explotar les possibilitats dels ***word embeddings***; també s'aplica a la traducció automàtica, els assistents tipus Siri, Google Assistant o Alexa, el *sentiment analysis*, la generació de resums o la combinació del PLN amb el reconeixement automàtic d'imatges.

Bibliografia

T. Ganegedara (2018). *Natural Language Processing with TensorFlow*. Packt Publishing.

3.5.6. Detecció de termes rellevants amb Word2Vec

Henry, Beth, Anne i Joseph decideixen fer un model dels titulars TOP amb el mètode Word2Vec. Gràcies a aquest model, podran conèixer el terme *Trump* i les seves companyies. Creuen que si traguessin el terme *Trump* com d'un cistell de cireres, arrossegarien els termes més significatius dels titulars per tenir contextos semblants. Assumeixen que els vectors *embedded* d'aquests termes són a prop del de *Trump*.

En el *notebook* PLA-1. 3.5 es mostra la creació del model dels titulars del tipus TOP, l'entrenament amb ell i la visualització de la distància entre els vectors dels termes[6]. La representació dels vectors de nombroses dimensions és difícil de visualitzar.

[6]En el *notebook* PLA-1 3.5 s'utilitza l'algorisme de la llibreria Gensim per a detectar termes.

No obstant això, la visualització és possible gràcies a una tècnica anomenada *t-SNE* (*t-distributed stochastic neighbor embedding*). La t-SNE redueix les dimensionalitats, de manera que és possible visualitzar les dades en un espai de dues dimensions sense perdre significativament la informació continguda a l'espai originari de nombroses dimensions. Altres mètodes de reducció de dimensions per a fer visualitzacions és el PCA (*principal component analysis*).

La figura 7 mostra la llista de tuples amb els termes més propers a *Trump*[7], ordenats segons el seu valor de proximitat. El model Word2Vec s'ha fet amb termes que apareixen tres vegades com a mínim.

[7]Les variants sinònimes *Donald Trump* i *Mr. Trump* s'han aglutinat en el terme *Trump*.

Figura 8. Termes més propers a *Trump* segons el model Word2Vec dels titulars POP

[('hold', 0.4118765470918369), ('c.e.o', 0.39993705465232043), ('deputy', 0.3538065093214791),
('win', 0.348140536254552), ('america', 0.34293782846949183), ('school',
0.32951093072057625), ('u.s', 0.3212339552410002), ('sex', 0.320808788112185), ('show',
0.30992592790911333), ('team', 0.30042379098722555), ('bolton', 0.2957814267552699),
('woman', 0.295501082936443), ('face', 0.2901126359692892), ('left', 0.2900806517515107),
('debate', 0.28658889897992407), ('ask', 0.2862524160673383), ('get', 0.2801972471163876),
('picture', 0.276551496695057), ('good', 0.2734423779977755), ('charge',
0.2693425141084042 6), ('lawyer', 0.2679939352346472), ('control', 0.25379695221250415),
('state', 0.24705133456763104), ('trade', 0.2460686002225894), ('race', 0.24421519481754128),
('chief', 0.24408938658380933), ('job', 0.24179929360434121), ('go', 0.239144779273626), ('gun',
0.2365317337489964), ('voter', 0.2348461031520803), ('gun control', 0.23393609899021056),
('facebook', 0.23159944668923804), ('fire', 0.2253085323505727), ('data',
0.21968546009527112), ('join', 0.21903136354333852), ('teacher', 0.21456841760683254), ('say',
0.2120758062710082), ('kushner', 0.2114884766218698), ('fear', 0.21105424485075885), ('stop',
0.21101526085876493), ('take', 0.20955965022835402), ('end', 0.2081539031478748), ('sue',
0.20813092218350437), ('power', 0.20645384934277566), ('plan', 0.20359681167356578),
('problem', 0.2034988668147314), ('college', 0.19522967924544055), ('silence',
0.19045767711048026), ('play', 0.18604973916620057), ('war', 0.18526196017560254), ('target',
0.18514794611908938), ('man', 0.18208618865480622), ('democrat', 0.18025277757698877),
('class', 0.17508857757215443), ('next', 0.1741021257106088), ('call', 0.1613580067020598),
('sign', 0.155942888042117), ('lead', 0.14944439981420052), ('void', 0.1492531471166409),
('new', 0.14164171749431204), ('aide', 0.1414166930866753), ('russia', 0.13515511290878007),
('age', 0.13507114225105687), ('house', 0.13422658386597502), ('porn star',
0.13036370042078124), ('abuse', 0.1236985572061892), ('president', 0.11380626723687277),
('make', 0.11347772784779273), ('trade war', 0.11213267626704823), ('tariff',
0.1032503878981752), ('prison', 0.09837882059707678), ('knew', 0.0881065073912421), ('talk',
0.08054224554315702), ('security', 0.05333730796407229), ('grow', 0.0531161619974365),
('mueller', 0.04004909702423757), ('g.o.p', 0.04004879000162311), ('firm',
0.03684471214451898), ('point', 0.005291083964170051), ('chaos', 0.0018369926847929818)]

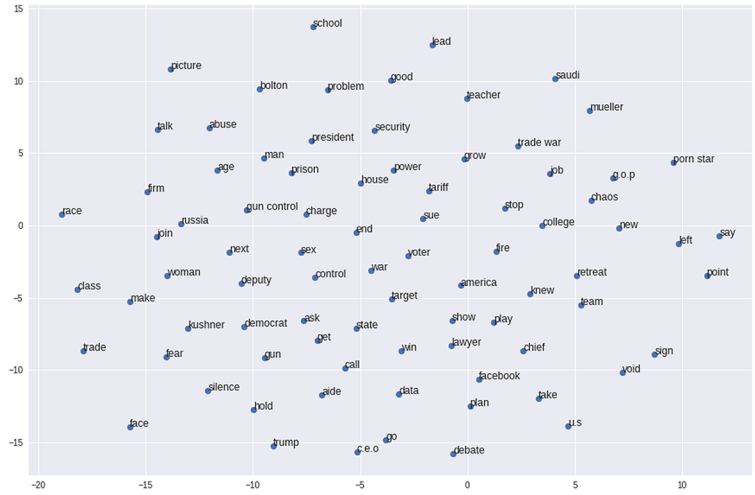
La figura 8 mostra la distància entre els termes segons el model Word2Vec dels titulars TOP.

Els resultats obtinguts es poden interpretar de diferents maneres. Henry, mirant els termes propers a Trump, destaca aquestes relacions en els titulars TOP:

- Les armes i el seu control als centres educatius ('gun control', 'school', 'college')
- El comerç i la guerra comercial ('trade', 'tariff', 'trade war')
- Escàndols com el de la relació amb una actriu pornogràfica ('sex', 'porn star') i l'ús fraudulent de dades per a la seva campanya electoral ('data', 'facebook')

Però Beth, mirant la gràfica de la figura 9, es fixa més en el fet que els termes del model dels titulars TOP estan bastant relacionats amb la violència ('gun', 'war', 'trade war') i altres aspectes negatius com 'problem', 'chaos', 'fear', 'fire', 'abuse' i accions judicials, que impliquen la presència de conflictes ('sue').

Figura 9. Distància dels termes amb una freqüència mínima de 3 en els titulars TOP segons Word2Vec



Ara bé, Beth considera que el model Word2Vec dels titulars TOP encara és dispers i confús per la presència de termes com *go*, *say* o *get*, que aporten molt poc com a termes motivadors de comentaris.

4. Detecció de temes (*topic detection*)

Beth vol tenir una visió més global i coherent dels termes agrupant-los per temes (*topics*). Segons ella, és esperable que l'agrupació per *topics* permeti identificar els titulars TOP amb un criteri més general i objectiu.

4.1. Detecció de temes amb Wordnet

Per a detectar els temes, Beth pensa a aprofitar la base de dades lèxica Wordnet de l'anglès. Wordnet recull el lèxic de la llengua anglesa segons una ontologia en la qual els sentits de les paraules es relacionen amb altres sentits per relacions d'hiperonímia i sinonímia (vegeu els apartats 3.3.2 i 3.3.3).

L'ontologia de Wordnet és un graf acíclic i dirigit (DAG). Cada node del graf és l'identificador d'un sentit, o *synset*, representat per un conjunt de paraules sinònimes, i cada arc v → w representa una relació que es verbalitza com «el *synset w* és un hiperònim del *synset v*» o com «el *synset v* és un hipònim del *synset w*».

Beth pensa que precisament aquesta relació d'hiperonímia és adequada per a agrupar paraules segons un criteri més lògic i abstracte, quelcom que ja li rondava pel cap quan volia aprofitar la relació d'hiperonímia per a normalitzar termes (c.f. 3.3.3). Si es parla, per exemple, de *Corea* i també de *Rússia*, es poden agrupar les dues paraules pel seu hiperònim, *país*. Així es podrà dir que els temes dels titulars no parlen tant de Corea i Rússia com de països en general.

En un graf es pot calcular la distància semàntica entre sentits segons els nodes que cal travessar recorrent els arcs que condueixen d'un sentit S1 a un sentit S2. Les paraules que són hipònims directes d'un *synset* són semànticament més properes que les paraules que tenen un hiperònim comú molt allunyat en el graf.

> Per exemple, *car*, *van*, etc. són més properes, perquè són hipònims més directes de *vehicle*, que *car* i *knife*, que tenen *artifact* com a hiperònim comú.

La llibreria NLTK permet calcular la distància entre dos sentits de Wordnet aplicant diferents mètriques. Una d'aquestes mètriques és la *Wu-Palmer Similarity*, amb un valor que va del 0 a l'1. Com més proper és a l'1 més propers són els sentits, i com més proper és al 0 més allunyats són els sentits.

Bibliografia

Vegeu l'apèndix C.3 del llibre de **Jurafsik** i **Martin** (2018) i http://bit.ly/2x6tplc, i http://www.nltk.org/howto/wordnet.html i http://www.nltk.org/howto/wordnet.html.

En el *notebook* PLA-1 4.1 ensenyem l'aplicació de la *Wu-Palmer Similarity* per a calcular la distància entre *dog* i *cat* i entre *United States* i *Spain*, en els sentits més habituals (indicats com a .01), quan la seva PoS és un nom (n).

Exemple de càlcul de distància entre dues *synsets* de Wordnet

LA DISTÀNCIA SEMÀNTICA ENTRE 'DOG' I 'CAT' ÉS 0.8571

LA DISTÀNCIA SEMÀNTICA ENTRE 'UNITED STATES' I 'SPAIN' ÉS 0.8

Beth pensa fer el *topic detection* creant un vector per a cada terme del vocabulari dels titulars. Les dimensions del vector recolliran les distàncies dels *synsets* del terme *t* respecte als *synsets* de tots els termes t_i del vocabulari de titulars. Una vegada se situen els vectors de cada terme en un espai vectorial, un algorisme de *clustering* agrupa els termes semànticament propers i es poden identificar els temes.

Beth aplica aquest mètode per justificar, per la seva proximitat semàntica, la unificació de *fire* i *attacks* en el titular següent sobre Trump (vegeu PLA-1 4), la qual cosa justificaria així mateix que els titulars prenen com a tema la intolerància del president.

> The "Daily Show" host said the attacks and fire of president Trump on fellow Republicans at a meeting reminded him of "a drunk uncle calling everyone out at a wedding".

Els *synsets* de *fire* i *attack*, quan són un nom, formen el vocabulari de *synsets*, que és el següent:

[Synset('attack.n.01'), Synset('attack.n.02'), Synset('fire.n.09'), Synset('approach.n.01'), Synset('attack.n.05'), Synset('attack.n.06'), Synset('attack.n.07'), Synset('attack.n.08'), Synset('attack.n.09'), Synset('fire.n.01'), Synset('fire.n.02'), Synset('fire.n.03'), Synset('fire.n.04'), Synset('fire.n.05'), Synset('ardor.n.03'), Synset('fire.n.07'), Synset('fire.n.08'), Synset('fire.n.09')]

Es genera el vector per cada *synset* del vocabulari. Per exemple, per a crear el vector de Synset('attack.n.01'), es crea una matriu que recull la distància entre aquest *synset* i cadascun dels *synsets* del vocabulari (vegeu la taula 17).

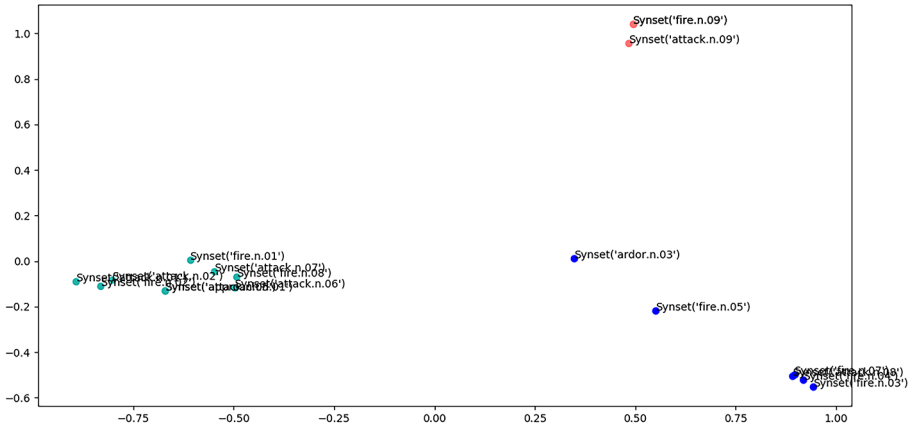
Taula 17. Matriu sobre la qual es crea el vector d'attack.n.01

attack. n.01	attack. n.02	fire. n.09'	approach. n.01	attack. n.05	attack. n.06'	attack. n.07	attack. n.08	attack. n.09	'fire. n.01'	fire. n.02	fire. n.03'	fire. n.04'	fire. n.05	ardor. n.03	fire. n.07	fire. n.08	fire. n.09
1.0	0.75	0.26	0.66	0.66	0.55	0.5	0.15	0.26	0.57	0.94	0.11	0.10	0.26	0.26	0.14	0.47	0.26

El vector creat a partir d'aquesta matriu, amb els vectors dels altres *synsets*, es posen en un espai vectorial i s'agrupen en tres clústers amb el mètode K-means, tal com es mostra en la figura següent. La figura evidencia que la hipòtesi de Beth és correcta, ja que fire.n.09, que Wordnet defineix com a *intense adverse criticism*, forma un clúster amb attack.n.09, que es defineix com a *strong criticism*[8].

(8)Les definicions dels *synsets* de Wordnet es denominen *glosses*.

Figura 10. Agrupació en clústers dels *synsets* de *fire* i *attack*



Henry exposa els seus dubtes sobre la conveniència d'aquest mètode de *topic detection* per a l'objectiu que persegueixen. Wordnet és un recurs que s'ha d'actualitzar amb nous termes.

Per exemple, *Trump* amb el *synset* de *president d'un país* no hi és. En canvi, *Kennedy* sí.

4.2. Wordnet, DBpedia i ConceptNet

Beth suggereix afegir la **DBpedia** com un recurs lèxic per al seu mètode. DBpedia organitza la Wikipedia com una base de dades oberta, gratuïta, amb una gran comunitat que la manté actualitzada. La finalitat de la base de dades és poder fer preguntes sobre fets i entitats i desenvolupar projectes de processament de llenguatge natural.

Consultant la DBpedia, es pot tenir la relació entre una entrada de la Wikipedia i un *synset* de Wordnet. Això és possible gràcies al fet que la DBpedia incorpora informació d'una altra ontologia, YAGO, que és l'acrònim de *yet another great onthology*. YAGO és una ontologia desenvolupada pel Max Planck Institute for Informatics, de Saarbrücken, i un dels seus objectius és relacionar la Wikipedia amb altres ontologies, entre les quals hi ha Wordnet.

La relació entre la Wikipedia i Wordnet, declarada en la DBpedia, ens permet conèixer el *synset* de Wordnet, que, segons YAGO, correspon a l'entrada *Donald Trump* de la Wikipedia. Curiosament, Donald Trump és considerat com una *celebrity* (vegeu PLA-1 4).

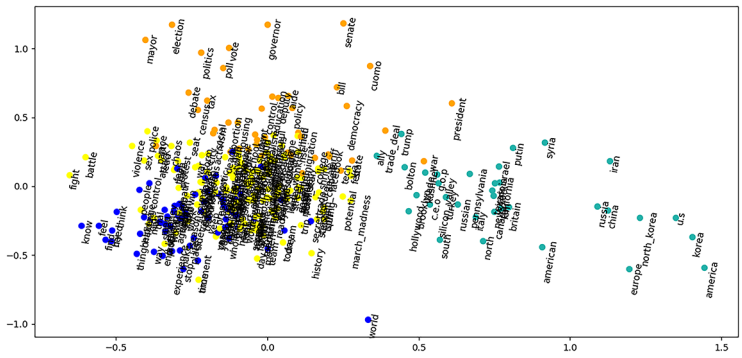
No obstant això, Henry no acaba de veure-ho clar. Els temes de molts titulars relacionen termes ontològicament molt distants, com per exemple *Trump*, *scorn* i *FBI*. Ell té la intuïció que, encara que es pugui dir que Trump provoca molts comentaris perquè és una *celebrity*, una característica dels titulars de les notícies més comentades és la relació de paraules distants ontològicament.

Henry i Beth sí que estan d'acord a considerar que els titulars, com qualsevol altre text, s'interpreten gràcies a unes relacions entre paraules que es consideren de sentit comú però que escapen de la formalització taxonòmica de Wordnet. Henry li parla de **ConceptNet**, una xarxa semàntica que és el fruit del projecte col·laboratiu *Open Mind Common Sense*, que es va iniciar l'any 1999 al MIT Media Lab. Aquesta xarxa semàntica pretén representar les relacions entre paraules segons el sentit comú dels parlants. Per a això, a més de fonts de coneixement lèxic (Wordnet, Wiktionary, DBpedia), afegeixen coneixement de relacions de sentit comú que van més enllà d'aquestes fonts i que s'han obtingut amb jocs d'associacions de paraules com Verbosity, dins del **GWAP** project.

Les diferències entre Wordnet i ConceptNet es manifesten en el càlcul de la relació semàntica entre termes. En el *notebook* PLA-1 4 podem veure el càlcul de la relació semàntica de *United States* i *White House* segons Wordnet i ConceptNet. Segons Wordnet, quan *White House* té el sentit de departament executiu del Govern dels Estats Units, el càlcul basat en les relacions taxonòmiques entre les dues entitats llança un valor molt petit (0,086), mentre que el resultat del càlcul de proximitat segons ConceptNet és 0,336, que és més conseqüent amb el sentit comú.

Beth s'anima per la possibilitat que ConceptNet ofereixi resultats de distància semàntica més propers al sentit comú. Per això decideix fer clusterització dels termes vectoritzant cada terme amb el càlcul de la seva distància semàntica respecte a la resta de termes. El recurs emprat per a calcular les distàncies és l'API de ConceptNet. En la figura següent es pot veure el resultat.

Figura 11. Agrupació en clústers dels termes dels titulars amb ConceptNet



La gràfica mostra coses interessants. D'una banda, es distingeix bastant bé un clúster que recull els termes de països i altres termes relatius a la política exterior (*Syria, China, North Korea, Putin*, etc.). D'altra banda, es perfila bastant bé un clúster de termes més relacionats amb la política i el funcionament democràtic (*democracy, senate, poll, vote, governor*, etc.), i en la frontera entre tots dos hi ha *Trump*.

Les altres classes són més confuses, encara que a Beth no se li escapa el fet que termes com *fight*, *battle*, *sex* o *violence* destaquin del seu grup.

4.3. LDA

Henry prefereix saber quins temes es tracten amb els mateixos articles sense utilitzar cap ontologia ni recurs extern. Per això aplica un dels mètodes no supervisats de detecció de temes més utilitzats: el ***latent dirichlet allocation*** o LDA.

El mètode LDA es basa en la idea que en un document es tracten temes (*topics*) diferents. L'LDA fa el model d'un document com una distribució de temes. Per a Henry, aquest model és més realista perquè en els titulars es relacionen temes heterogenis, la política amb el món de l'espectacle, etc.

Presentem com es fa *la topic detection*:

1) Decidir un nombre *K* de temes. Després es pot anar afinant el nombre segons els resultats.

2) Per a cada document, assignar a cada paraula un dels *K* temes de manera aleatòria. D'aquesta manera, tenim la representació dels temes de tots els documents i la distribució de les paraules en aquests temes (encara que no és la distribució òptima).

3) Per a millorar-ho, per a cada document *d*:

a) Anar a cada terme per actualitzar l'assignació d'un tema segons dos criteris:

- Quant prevalent és el terme en tots els temes.
- Quant prevalents són els temes en el document *d*.

Els dos criteris

Els dos criteris són explicats més extensament en http://bit.ly/2x6knum.

4) Repetir aquest procés de millora per a cada paraula en tots els documents passant per tota la col·lecció de documents moltes vegades. Aquesta actualització iterativa i múltiple assegura que al final s'obtingui una distinció coherent de temes.

4.4. Recursos per a fer *topic detection*

Ldatuning és una llibreria en R que aplica l'LDA per fer *topic detection*. En Python és possible fer-ho també amb la llibreria **Sklearn**, encara que és recomanable **pyLDAvis** si es vol fer una presentació clara i atractiva dels temes i la contribució dels termes en la detecció d'aquests temes.

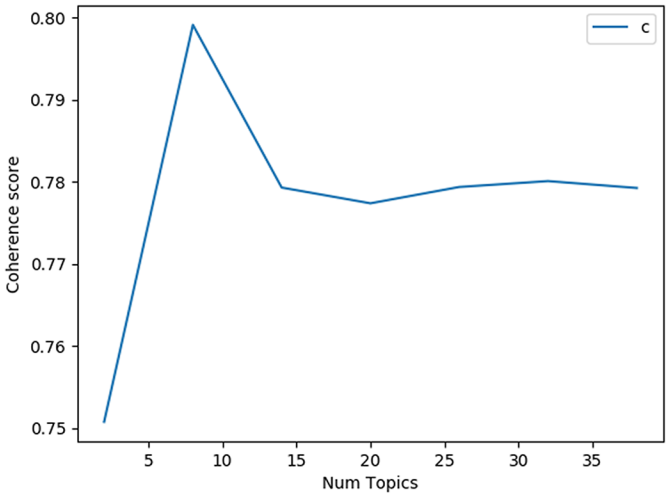
4.5. Resultats

A Henry i a Joseph els interessa aplicar l'LDA. A Henry li interessa per les raons que ja hem explicat, i a Joseph perquè creu que la contribució d'un terme al tema pot ser útil per a entrenar un predictor de notícies provocadores de comentaris. Moguts per aquest interès, calculen la distribució dels termes[9] dels titulars TOP per temes. Per suggeriment de Beth, es concentren en termes nominals, ja que els temes d'un text són més fàcils d'identificar amb els seus sintagmes nominals.

[9] Els termes tenen una freqüència mínima de 3, com en el model Word2Vec.

Abans de començar, decideixen que cada titular serà un document, però han de decidir el nombre de temes, això és, la *K*. Com que no es pot saber *a priori* el nombre de temes sobre els quals tracten els titulars, es fa un càlcul del nombre de *K* aproximat a partir del qual augmentar-ne el nombre no comporta obtenir resultats coherents amb el contingut dels titulars. En la figura 12 es mostra que el nombre de *K* més conseqüent amb els titulars és prop de 7.

Figura 12. Resultat del càlcul de la *K* més coherent per a fer un LDA dels titulars TOP



Les columnes de la figura 13 mostren les deu paraules més probables de pertànyer a un tema, ordenades per valor de probabilitat.

Figura 13. Termes dels titulars TOP ordenats segons la seva probabilitat de pertànyer a un tema

	1	2	3	4	5	6	7
0	"trump"	"facebook"	"team"	"school"	"get"	"u.s"	"tariff"
1	"trade"	"state"	"aide"	"picture"	"trade war"	"america"	"say"
2	"talk"	"college"	"bolton"	"war"	"problem"	"lawyer"	"plan"
3	"chaos"	"sue"	"target"	"gun"	"control"	"security"	"russia"
4	"point"	"go"	"race"	"democrat"	"face"	"sex"	"gun control"
5	"g.o.p"	"abuse"	"mueller"	"good"	"house"	"lead"	"teacher"
6	"voter"	"call"	"saudi"	"win"	"c.e.o"	"play"	"void"
7	"fear"	"data"	"president"	"age"	"charge"	"power"	"sign"
8	"job"	"firm"	"class"	"left"	"job"	"woman"	"join"
9	"show"	"stop"	"fire"			"president"	

Veiem que les paraules més probables de pertànyer a un tema no defineixen temes distintius. Més aviat, tornem a veure els que havíem vist relacionats amb Trump, com el control d'armes, la guerra comercial, els escàndols de Facebook i els sexuals. Són termes repartits pels temes sense perfilar un camp temàtic recognoscible que no sigui el dels assumptes de Trump. Sembla, per tant, que els temes graviten entorn de la figura del president.

De totes maneres, a Beth li continua cridant l'atenció que, entre els termes que per si mateixos no apunten a un tema concret, dominin paraules relacionades amb la violència i el poder (*war*, *power*) o bé que tinguin connotacions negatives (*problem, fear, chaos, abuse*).

5. Predicció

El grup es reuneix amb Peter, a qui sorprenen els resultats obtinguts amb la detecció de temes. Veu que *Trump* és el terme principal en els titulars del tipus TOP i li sorprenen certes agrupacions que s'han trobat en els temes (p. ex., *war* i *pornstar* o *control* i *sex*).

Fan recopilació de les dades que han anat obtenint:

1) El lema i la PoS dels termes dels titulars.

2) El tf.idf dels termes en els titulars TOP.

3) La proximitat semàntica entre els termes dels titulars TOP.

4) Els temes als quals pertanyen els termes dels titulars TOP.

Peter pregunta si amb aquestes dades ja seria possible fer un predictor de notícies motivadores de comentaris, que és, de fet, l'objectiu últim del projecte. Joseph diu que els titulars ja s'han etiquetat com a TOP i NOTOP (vegeu l'apartat 3.4.2), amb la qual cosa és possible abordar la predicció aplicant aprenentatge automàtic. L'objectiu seria que el predictor aprengués la tasca de classificar titulars TOP i NOTOP.

Peter els diu que posin mà a l'obra.

5.1. Passos que cal fer

El mètode per a crear el predictor té els passos següents:

1) Preprocessament dels titulars fins a tenir les dades amb les quals el classificador aprendrà a classificar els titulars.

2) Entrenament del predictor.

3) Una vegada entrenat, etiquetatge de titulars nous pel predictor.

5.1.1. Preprocessament

El preprocessament consisteix a:

1) Associar els documents amb la classe que el classificador ha de distingir.
2) Netejar els documents de caràcters estranys i de *tokens* que no interessen per a fer l'anàlisi (URL, noms propis, emoticones, etc.).

3) Preparar les dades que el classificador ha d'aprendre.
4) Preparar un corpus d'entrenament i de test.

1) Associar els documents amb la seva classe. Els documents han d'estar etiquetats segons la classe a la qual pertanyen i que el classificador ha d'aprendre a identificar. En el cas que ens ocupa, les classes són dues: TOP i NOTOP. Els documents TOP són els titulars de notícies que han motivat un nombre de comentaris que superen un llindar. Els documents NOTOP són la resta de titulars. L'associació es fa assignant als titulars una etiqueta (*data_labelling*), que denota la classe a la qual pertanyen.

2) Netejar el corpus. La neteja del corpus no comporta molt d'esforç en el cas dels titulars del *New York Times*. Els titulars són tal com van aparèixer publicats en el diari i, per tant, no cal eliminar caràcters especials ni emoticones, corregir faltes d'ortografia, eliminar dobles espais, etc. El cas seria molt diferent si s'haguessin de preprocessar tuits, correus electrònics, fòrums i altres documents d'escriptura informal, descurada, amb errors de tecleig i amb una mescla de caràcters especials en els emojis amb caràcters alfanumèrics.

3) Preparar les dades. La preparació de les dades es fa amb un vectoritzador. Per al cas que ens ocupa, Beth, Henry, Joseph i Anne decideixen utilitzar les *features* d'un *tf.idf vectorizer*. El vectoritzador obté les *features* dels titulars amb els procediments ja explicats en l'apartat 3.4.

4) Preparar un corpus d'entrenament i de test. El grup decideix preparar el 80% del total de titulars TOP i NOTOP per a l'entrenament i prepara el 20% restant per avaluar el classificador (corpus de test). Es distingeixen, tant per al corpus d'entrenament com per al de test, les anomenades *dades X*, que són els vectors que descriuen els documents, i les *dades Y* (o *target*), que són les etiquetes associades als vectors, això és, TOP o NOTOP. A continuació veiem un exemple de dades *X* i *Y*.

Taula 18. Exemples de dades *X* i dades *Y*

Text	Dades *X* (Vectorització del text amb un vectoritzador tf.idf)	Dades *Y*
Titular 1	[0, 0, 0.78, 0.23,....]	TOP
Titular 2	[0, 0.45, 0.01, 0.26,....]	NOTOP

5.1.2. Entrenament

L'entrenament consisteix a configurar un model dels titulars. Hi ha un bon nombre d'algorismes per a configurar aquest model, i alguns són més adequats que d'altres per a classificar certs fenòmens. És normal avaluar els resultats de la predicció segons un model i comparar els resultats obtinguts amb els que s'obtenen amb models fets amb altres algorismes.

De moment, el grup de Beth, Henry, Joseph i Anne configuren un model basat en el *logistic regression model*, també conegut com *logit regression, maximum-entropy classification* o *log-linear classifier*. Aquest model és usat normalment per a classificacions amb dos valors possibles, com és el cas de TOP i NOTOP. És similar a la *linear regression*, amb la diferència que les dades *Y* no són contínues, sinó discretes, referides a una classe. A més, la *regression model*, al contrari que la *linear regression*, permet estimar la probabilitat que un titular pertanyi a una de les classes.

El classificador s'entrena ajustant les dades (*fit*) del corpus d'entrenament al model establert.

5.1.3. Predicció

Una vegada entrenat, el classificador prediu les etiquetes de les dades *X* del corpus de test. Això és, a partir de les dades *X* prediu les dades *Y*. A partir d'aquestes prediccions, s'avalua el classificador en funció de com s'allunyen les prediccions respecte a les assignacions d'etiqueta que són en el corpus de test.

També es poden saber les *features* que han estat més informatives a l'hora de classificar els titulars. Les *features* més informatives ens donen pistes sobre la qualitat del classificador i ens poden oferir informació interessant.

5.1.4. Recursos per a fer la predicció

La predicció es pot fer amb la llibreria de Python NLTK, que importa els classificadors de la llibreria Scikit-learn. R té llibreries ja carregades que permeten fer models amb *linear regression* i *logistic regression*, per exemple. D'altra banda, Apache Spark ofereix API en Java, Scala, Python i R per a fer models de classificació.

5.2. Resultats

De moment presentem els resultats obtinguts per Beth, Henry, Joseph i Anne sobre les *features* més informatives per a fer la classificació (vegeu el *notebook* PLA-1, 5).

Figura 14. Informativitat dels termes en la classificació de titulars TOP i NOTOP

NONTOP	-0.8058405273727582	episode
NONTOP	-0.7534346524087436	black
NONTOP	-0.7073540060974302	season
...		
TOP	**3.6327081228639**	**trump**
TOP	1.2363090722630365	picture
TOP	1.1256915185912726	president
TOP	1.0921666744283134	job
TOP	0.9698316413604575	voter
TOP	0.9176627731358946	school
TOP	0.9136615761705841	bolton
TOP	0.9098591448319565	chaos
TOP	0.8632258255927439	college
TOP 0.8446625962106591 trade		

Segons aquests resultats, s'evidencia una vegada més la importància de *Trump* a l'hora de predir els titulars amb més comentaris. D'altra banda, en comparar els valors d'informativitat dels titulars etiquetats com a TOP amb els valors molt inferiors dels titulars NONTOP, es confirma que les *features* més informatives caracteritzen sobretot els titulars de notícies que motiven més comentaris.

De sobte, Peter s'emociona. Per un moment ha vist passar el temible espectre de l'*overfitting*. El fet que el terme més informatiu sigui *Trump*, que és una referència molt puntual a una persona, pot ser un avís que hagin fet un classificador molt bo per a predir les etiquetes dels titulars amb els quals s'ha entrenat però que sigui dolent per a predir les etiquetes de titulars nous. En l'avaluació del sistema caldrà comprovar que no es produeix *overfitting*.

No obstant això, Peter i el seu equip reconeixen que el sistema serà segurament un classificador *one-shot* dependent dels canvis en el temps, les tendències i els temes d'actualitat. Peter es pregunta: «I, quan no hi hagi Trump, servirà el nostre classificador?» Els membres del seu equip, excepte Beth, s'encongeixen d'espatlles: «Caldrà actualitzar-lo cada *X* temps», diuen. No obstant això, Beth pensa que és possible fer un classificador al qual no afecti què o qui és notícia en un moment concret.

Resum

En aquest mòdul hem introduït els mètodes de processament de llenguatge natural més comuns per a processar informació textual i obtenir dades per a prendre una decisió. La presentació d'aquests mètodes ha tingut un fil argumental, que ha estat un projecte hipotètic per al *New York Times* que vol automatitzar la inclusió de publicitat en les notícies que provoquen molts comentaris. El diari subcontracta un equip de lingüistes computacionals per desenvolupar un prototip de predictor de notícies motivadores de comentaris.

Hem presentat mètodes considerats bàsics, com la detecció de termes i col·locacions, l'etiquetatge segons la categoria gramatical, la lematització i l'etiquetatge semàntic. Després hem explicat mètodes de transformació de les dades lingüístiques en representacions que faciliten el processament d'aquestes dades i l'obtenció d'informació rellevant. D'una banda, hem explicat la transformació dels documents i dels termes en vectors i, d'altra banda, hem explorat mètodes de càlcul per saber la rellevància dels termes en un document i la relació semàntica entre els termes d'un document. Ho hem fet comparant el càlcul de la relació semàntica en un espai vectorial amb la relació semàntica en una ontologia com Wordnet. També hem presentat un mètode d'agrupació de paraules per temes.

Finalment, hem presentat un mètode d'aprenentatge automàtic per a predir dades vinculades a les opinions, com predir si una notícia motivarà molts comentaris. La predicció s'ha fet amb un classificador de textos. El mètode de predicció ha motivat qüestions importants com el perill de l'*overfitting* i també la consideració del classificador com un sistema que s'ha d'actualitzar amb noves dades cada cert temps.

De moment, hem ensenyat un mètode de classificació. Ara bé, hi ha molts mètodes alternatius i caldrà avaluar els resultats obtinguts amb diversos per a triar el més adequat.