Tecnologías emergentes y datos abiertos:
PROCESAMIENTO DE LENGUAJE NATURAL

ACTION

Contenido elaborado por Alejandro Alija, experto en Transformación Digital y datos abiertos.

Abril 2020

ACTION

En la sección *Metodología*, introdujimos al lector sobre la forma en la que se estructura este informe. El *recorrido AIA* (*Awareness*, *Inspire*, *Action*) nos permite adentrarnos de forma gradual en el tema del Procesamiento del Lenguaje Natural, desde los conceptos más básicos hasta el desarrollo de un caso práctico indicado para aquellas personas que quieran pasar a la *Action.*

En esta sección hemos decidido desarrollar un ejemplo que nos permitirá analizar las principales métricas (cantidad de palabras de cada tipo, grupos de palabras que más se repiten, principales relaciones entre palabras, etc.) de los comentarios realizados por ciudadanos sobre determinados debates que se ponen de manifiesto en una plataforma web ciudadana. Con este ejemplo seremos capaces de **procesar los textos para extraer las principales palabras, tipos de palabras y el análisis de sentimiento.** Así, podremos determinar aquellos debates planteados por la ciudadanía que más preocupan o que más división generan. Sin herramientas que automaticen el análisis de este tipo de textos, los debates y las opiniones de la ciudadanía corren el riesgo de pasar desapercibidos puesto que no es viable que todas las opiniones sean analizadas por un ser humano.

El conjunto de datos

En este caso de uso utilizaremos un conjunto de datos disponible en el catálogo de datos de datos.gob.es. En particular utilizaremos la distribución de **Participación ciudadana. Debates y propuestas** accesibles desde el siguiente enlace: https://datos.gob.es/es/catalogo/l01280796-participacion-ciudadana-debates-y-propuestas1

Esta distribución contiene Información de debates y propuestas que figuran en la plataforma de participación ciudadana http://decide.madrid.es. Además, se incluyen comentarios y votaciones, así como la información auxiliar necesaria para entender el contenido de los datos. Hecha ya la introducción a nuestro ejercicio. ¡Comencemos!

Echar un vistazo a la web original desde donde se generan los datos que vamos a trabajar nos puede ayudar a entender la estructura del conjunto de datos. Por ello, nos asomamos a decide.madrid.es para entender cómo funciona la plataforma.

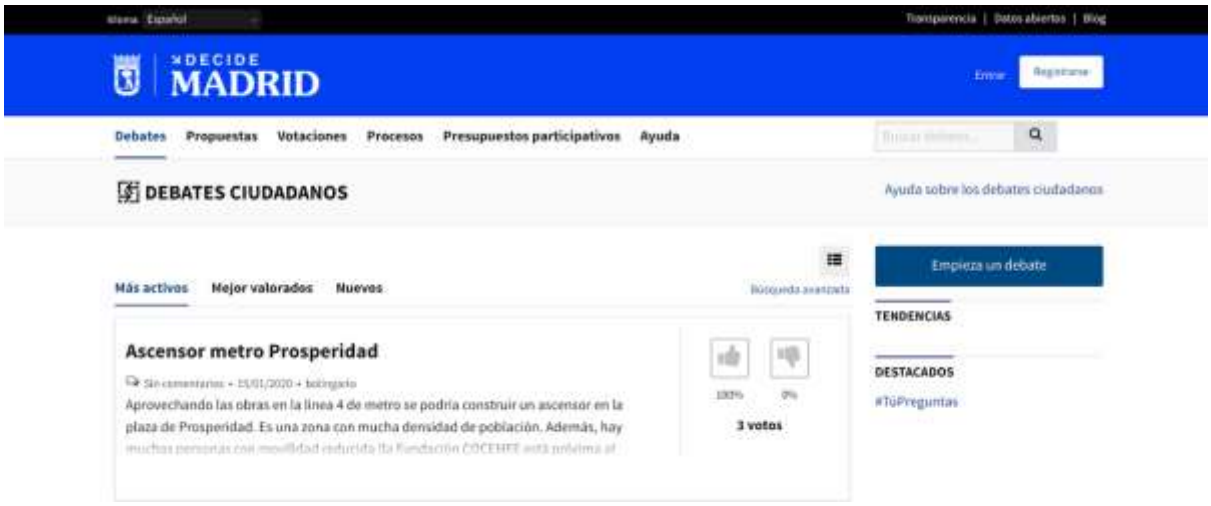

Figura 10. *Plataforma digital Madrid decide*. http://decide.madrid.es.

Si en vez de acceder a la interfaz de la web donde la ciudadanía interactúa visitamos el catálogo de datos.gob.es, nos encontramos con las siguientes distribuciones de datos accesibles.

Figura 11. *Vista del portal de datos abiertos datos.gob.es donde se encuentran los conjuntos de datos disponibles procedentes de la plataforma Madrid decide.*

Cuando descargamos una distribución disponible, por ejemplo, *Comentarios a debates y propuestas.csv* vemos que éste tiene el siguiente aspecto:

	id	commentable_id	commentable_type	body	created_at	cached_votes_total	cached
1	9	3	Debate	Ilusionante sin duda.	07/09/2015 10	9	7
2	10	3	Debate	Queda por ver cómo se consigue a...	07/09/2015 10	7	5
3	11	4	Debate	Estoy de acuerdo contigo Jesús, pe...	07/09/2015 10	2	1
4	14	3	Debate	Me parece muy buen paso, pero de...	07/09/2015 10	1	1
5	19	3	Debate	Hola, Fuencisla: ¿Qué canales alter...	07/09/2015 11	1	1
6	21	8	Debate	A mí las máquinas esas tampoco ...	07/09/2015 11	4	2
7	22	7	Debate	Creo que hay que tratar de prestar...	07/09/2015 11	2	2
8	26	9	Debate	sobre todo fuentes, en las que pod...	07/09/2015 11	4	4
9	28	10	Debate	Me parece muy buena medida. Co...	07/09/2015 11	3	2

Showing 1 to 10 of 125,450 entries, 10 total columns

Figura 12. *Previsualización del conjunto de datos con el contenido de los comentarios en los debates.*

Los campos que se incluyen en este fichero son:

```
'data.frame':    125450 obs. of  10 variables:
 $ id                : int  9 10 11 14 19 21 22 26 28 29 ...
 $ commentable_id    : int  3 3 4 3 3 8 7 9 10 9 ...
 $ commentable_type  : chr  "Debate" "Debate" "Debate" "Debate" ...
 $ body              : chr  "Ilusionante sin duda." "Queda por ver cómo se consigue
articular y qué tal se desenvuelve la gente, pero promete." "Estoy de acuerdo contigo
Jesús, pero si algún día se consigue solucionar el problema con las contratas y empeza"|
__truncated__ "Me parece muy buen paso, pero deja fuera a todas la personas que no tienen
acceso a internet o a personas mayor"| __truncated__ ...
 $ created_at        : chr  "07/09/2015 10" "07/09/2015 10" "07/09/2015 10" "07/09/2015
10" ...
 $ cached_votes_total: int  9 7 2 1 1 4 2 4 3 4 ...
 $ cached_votes_up   : int  7 5 1 1 1 2 2 4 2 4 ...
 $ cached_votes_down : int  2 2 1 0 0 2 0 0 1 0 ...
 $ ancestry          : chr  "" "" "" "" ...
 $ confidence_score  : int  388 214 0 100 100 0 200 400 66 400 ...
```

Sin entrar en todo el detalle,

- *id*
- *commentable_id*

son identificadores que sirven para relacionar este fichero con otros como ahora veremos.

- *commentable_type*

hace referencia al tipo de foro al que se refieren los comentarios, que pueden ser debates, propuestas o encuestas.

- *Body*

es el cuerpo de los comentarios ciudadanos.

A partir de ahí, el resto de campos no se usarán en este ejemplo.

Para entender por completo la situación, es necesario descargar también el fichero debates.csv. Este fichero contiene los identificadores y las descripciones de los debates sobre los cuales, luego, la ciudadanía hace sus comentarios que quedan recogidos en el fichero *Comentarios a debates y propuestas.csv.* Veamos un ejemplo de este otro fichero.

	id	title	description	created_at	cached_vo
1	3	¿Qué os parece este nuevo espacio de debate?	<p>Empezamos a abrir secciones con este espacio d...	06/09/2015 14	1570
2	4	Basuras Moncloa	<p>Ya sé que es un debate manido, pero el distrito e...	07/09/2015 10	57
3	5	Funciones de la policía municipal de Madrid.	<p>Propongo repensar algunas de las funci...	07/09/2015 10	250
4	7	Madrid ciclista y bicimad	<p>Me gustaría saber qué problemas detectamos qu...	07/09/2015 10	322
5	8	¿SOPLADORAS? NO GRACIAS	<p>La sopladoras son máquinas realmente infernales...	07/09/2015 10	876
6	9	Fuentes públicas, bancos y sombras	<p>Las calles y plazas de Madrid se han vuelto duras...	07/09/2015 11	3597
7	10	Madrid Río y la convivencia entre ciclistas y peatones	<p>Propongo realizar una adaptación a la r...	07/09/2015 11	600
8	12	Publicidad sexual en coches	<p>A nadie que tenga coche y no disponga de garag...	07/09/2015 11	645
9	13	Devolver ON29	<p>Propongo que se devuelva el solar de Ofelia Niet...	07/09/2015 11	68

Showing 1 to 10 of 3,723 entries, 10 total columns

Figura 13. *Previsualización del conjunto de datos que contiene la relación y descripción de los debates ciudadanos.*

Código y resultados

Una vez que tenemos los datos de entrada para analizar nuestro caso de uso, comencemos con el análisis. Este ejemplo ha sido realizado íntegramente bajo entorno de programación R. Se ha utilizado código R, el IDE de programación *RStudio* y el principal paquete para el análisis del lenguaje natural es *udpipe*.

Los comentarios embebidos en el código están en inglés siguiendo las buenas prácticas recomendadas en programación.

En este ejemplo hemos reducido la dimensionalidad del fichero original que contiene los comentarios de los debates. Hemos analizado 100 debates diferentes y 3.170 comentarios individuales en cuestión de segundos. Una cifra nada desdeñable si tuvieran que ser analizados por una persona.

```
#Lets start the analysis

data(debates_comentarios_filtered)

ud_model <- udpipe_download_model(language = "spanish")
ud_model <- udpipe_load_model(ud_model$file_model)
x <- udpipe_annotate(ud_model, x = debates_comentarios_filtered$body, doc_id
=debates_comentarios_filtered$commentable_id)
x <- as.data.frame(x)
```

El paquete *UDPipe* proporciona herramientas de *text encoding*, etiquetado y análisis de dependencia que se puede aplicar a textos sin procesamiento previo, y cubre una parte esencial en el Procesamiento del Lenguaje Natural. Para una información mucho más detallada del funcionamiento del paquete se puede consultar el artículo original *Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe*.

Una de las grandes ventajas de utilizar un paquete como este es la facilidad para utilizar modelos de lenguaje pre-entrenados, cómo introducimos en la sección *Cómo hacemos que las máquinas entiendan el lenguaje humano* donde explicamos las ventajas de los espacios de *word embeddings.* En este caso, como se ve en código, descargamos e incluimos en nuestro código un modelo pre-entrenado en español. *UDPipe* incluye modelos para más de 64 idiomas.

Una vez ejecutado las funciones *ud_model* y *udpipe_annotate* conseguimos convertir el fichero con los comentarios iniciales en el siguiente conjunto de datos que contiene un análisis del texto de los comentarios.

	doc_id	paragraph_id	sentence_id	sentence	token_id	token	lemma	upos	xpos	feats
1	1	1	1	ilusionante sin duda.	1	ilusionante	ilusionante	ADJ	NA	NA
2	1	1	1	ilusionante sin duda.	2	sin	sin	ADP	NA	NA
3	1	1	1	ilusionante sin duda.	3	duda	duda	NOUN	NA	Gender=Fem|Number=Sing
4	1	1	1	ilusionante sin duda.	4	.	.	PUNCT	NA	NA
5	1	2	1	Queda por ver cómo se consigue articular y qué tal se...	1	Queda	quedar	VERB	NA	Mood=Ind|Number=Sing|Person=3|Tense=Pres|Verb
6	1	2	1	Queda por ver cómo se consigue articular y qué tal se...	2	por	por	ADP	NA	NA
7	1	2	1	Queda por ver cómo se consigue articular y qué tal se...	3	ver	ver	VERB	NA	VerbForm=Inf
8	1	2	1	Queda por ver cómo se consigue articular y qué tal se...	4	cómo	cómo	ADV	NA	NA
9	1	2	1	Queda por ver cómo se consigue articular y qué tal se...	5	se	él	PRON	NA	Case=Acc,Dat|Person=3|PrepCase=Npr|PronType=Pr
10	1	2	1	Queda por ver cómo se consigue articular y qué tal se...	6	consigue	conseguir	VERB	NA	Mood=Ind|Number=Sing|Person=3|Tense=Pres|Verb

Figura 14. *Resultado del análisis del algoritmo utilizado por UDPipe para tokenizar y anotar el texto de los comentarios ciudadanos.*

De la misma forma que vimos en la introducción de Awareness los algoritmos de *UDPipe* separan cada palabra en las oraciones que forman los comentarios y les asigna un índice (aquí llamado *token_id*). Las avanzadas herramientas de *UDPipe* nos permiten clasificar las palabras por tipo en función de que sean nombres, adjetivos, signos de puntuación, etc.

Gracias a estas clasificaciones de palabras, cada vez estamos más cerca de que un programa *entienda* el significado de los comentarios hechos por las personas.

En la mayoría de los idiomas, los sustantivos (nombres) son los tipos de palabras más comunes, junto a los verbos. Sustantivos comunes y verbos son las palabras más relevantes para fines analíticos. Junto a éstos, los adjetivos y los nombres propios son las siguientes palabras más importantes en NLP.

Profundizando en la clasificación de palabras, este es el resultado que encontramos en nuestro análisis de comentarios.

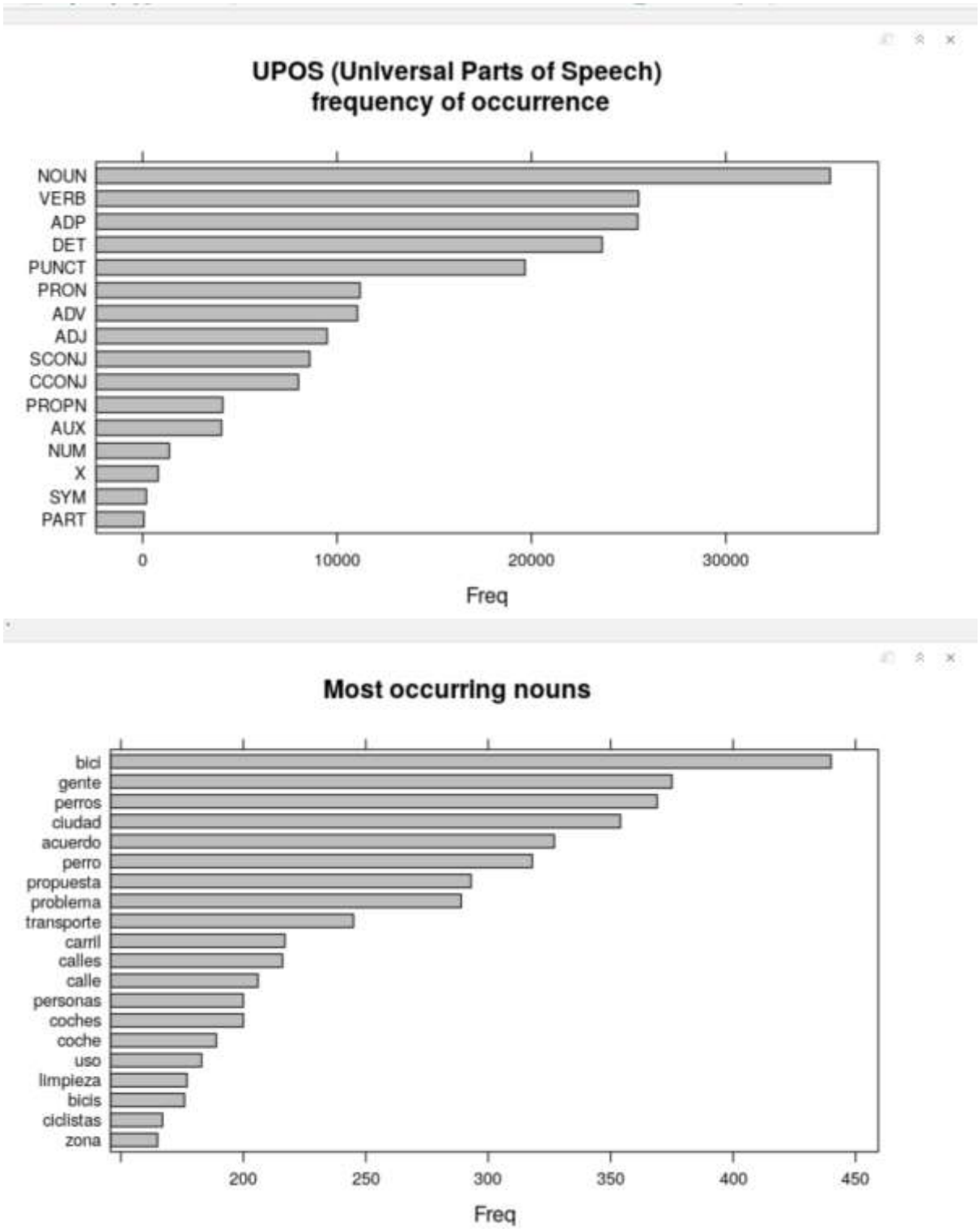

Figura 15. *Representación gráfica de algunos de los indicadores producidos por UDPipe. Panel superior: análisis UPOS (Universal Part of Speech) que indica los tipos de palabras más comunes en el conjunto de datos. Panel inferior: nombres más comunes que aparecen en el conjunto de datos.*

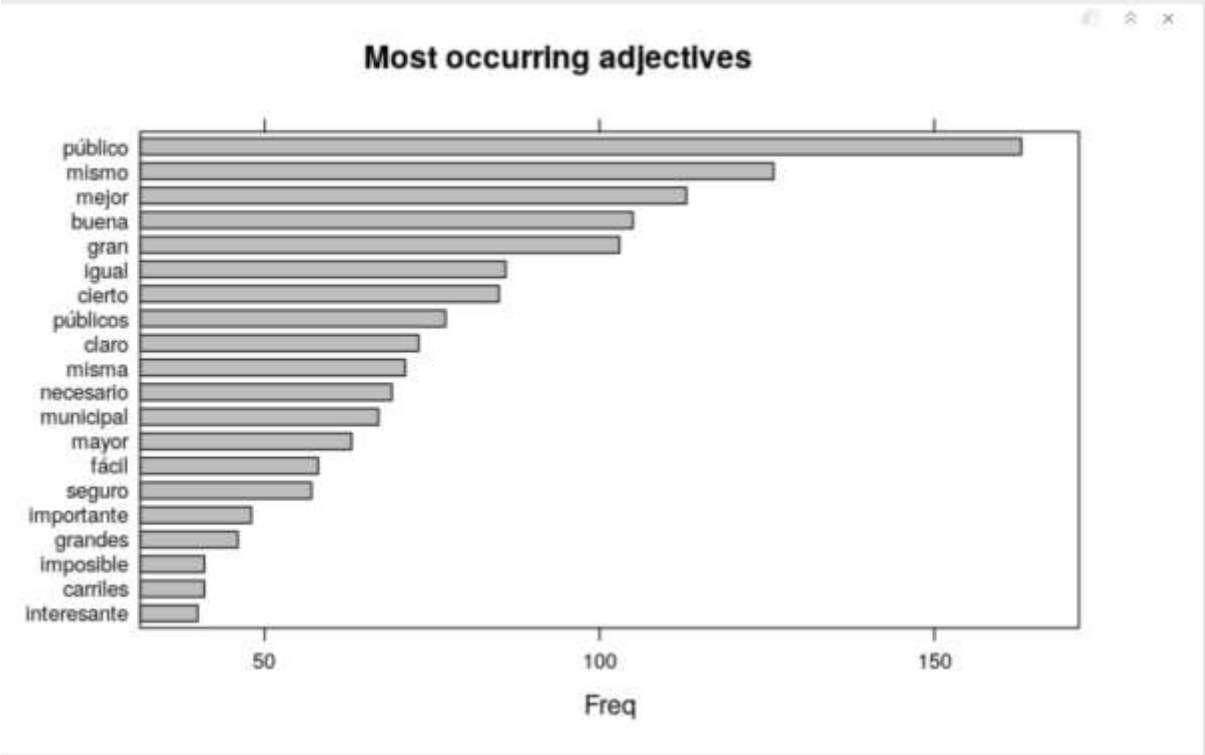

Figura 16. *Adjetivos más comunes en el conjunto de datos.*

Cómo vemos, bici, gente, perros, ciudad, etc. son los nombres más comunes en los comentarios. De la misma forma, público, mismo, mejor, buena, etc. son los adjetivos que más se repiten. ¡**Ya tenemos nuestro primer resultado importante!** Sabemos que la ciudad está hablando fundamentalmente de *estos temas.* Sin embargo, uno de los grandes desafíos del NLP es detectar las relaciones entre palabras. Es decir, para definir un tema no basta con identificar aquellos nombres que más se repiten. En lingüística, las estructuras que definen correctamente un tema son estructuras más complejas que simples palabras sueltas.

Utilizamos uno de los métodos (RAKE, Rapid Automatic Keyword Extraction) que viene con *UDPipe* para extraer combinaciones de palabras a modo de expresiones clave que la ciudadanía utiliza en sus comentarios.

```
## Using RAKE
stats <- keywords_rake(x = x, term = "lemma", group = "doc_id",
                       relevant = x$upos %in% c("NOUN", "ADJ"))
stats$key <- factor(stats$keyword, levels = rev(stats$keyword))
barchart(key ~ rake, data = head(subset(stats, freq > 3), 20), col = "grey",
         main = "Keywords identified by RAKE",
         xlab = "Rake")
```

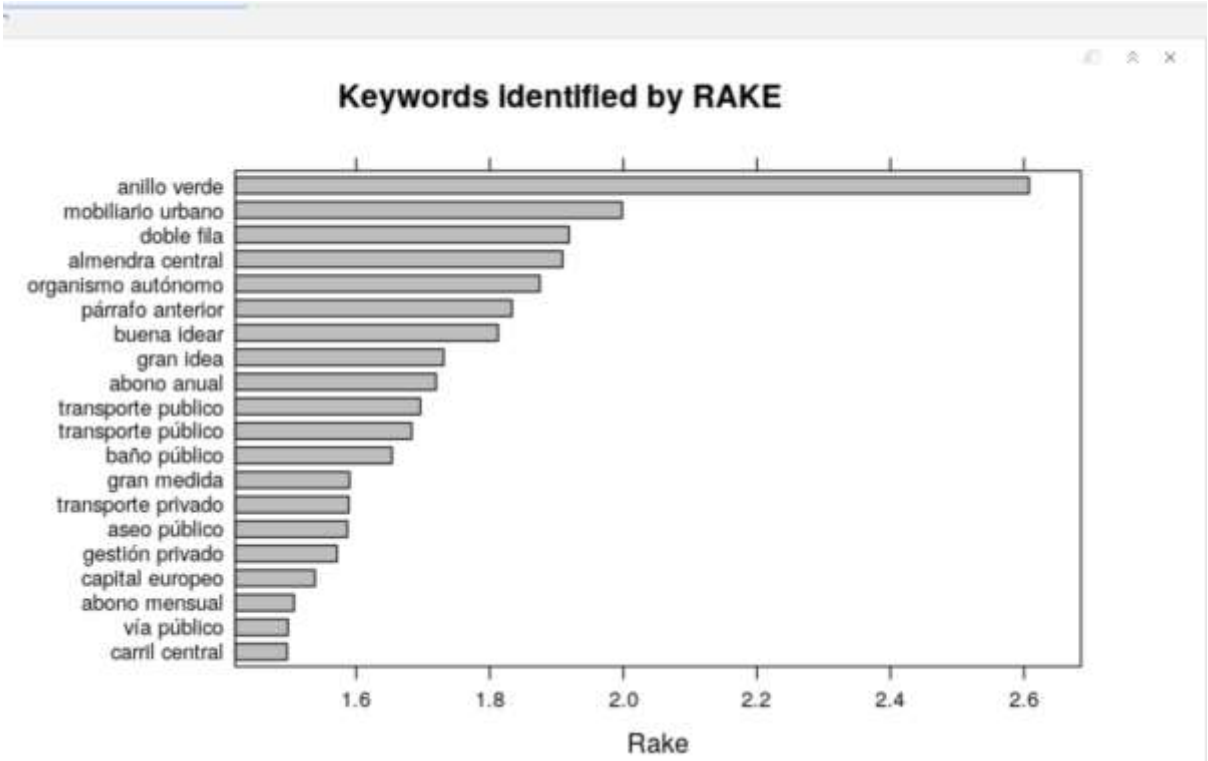

Figura 17. *Conjuntos de términos o expresiones clave más comunes en el conjunto de datos de comentarios.*

Ahora ya sabemos que la ciudadanía habla (por orden en el número de apariciones) sobre el anillo verde, el mobiliario urbano, los aparcamientos en doble fila, etc. Este es un resultado más valioso que la simple frecuencia de aparición de nombres y adjetivos.

Analicemos ahora otro aspecto muy importante en el NLP. Se trata del análisis de las coincidencias. Las coincidencias permiten ver cómo se usan las palabras en la misma oración o una al lado de la otra. Veamos los resultados a través de la siguiente visualización.

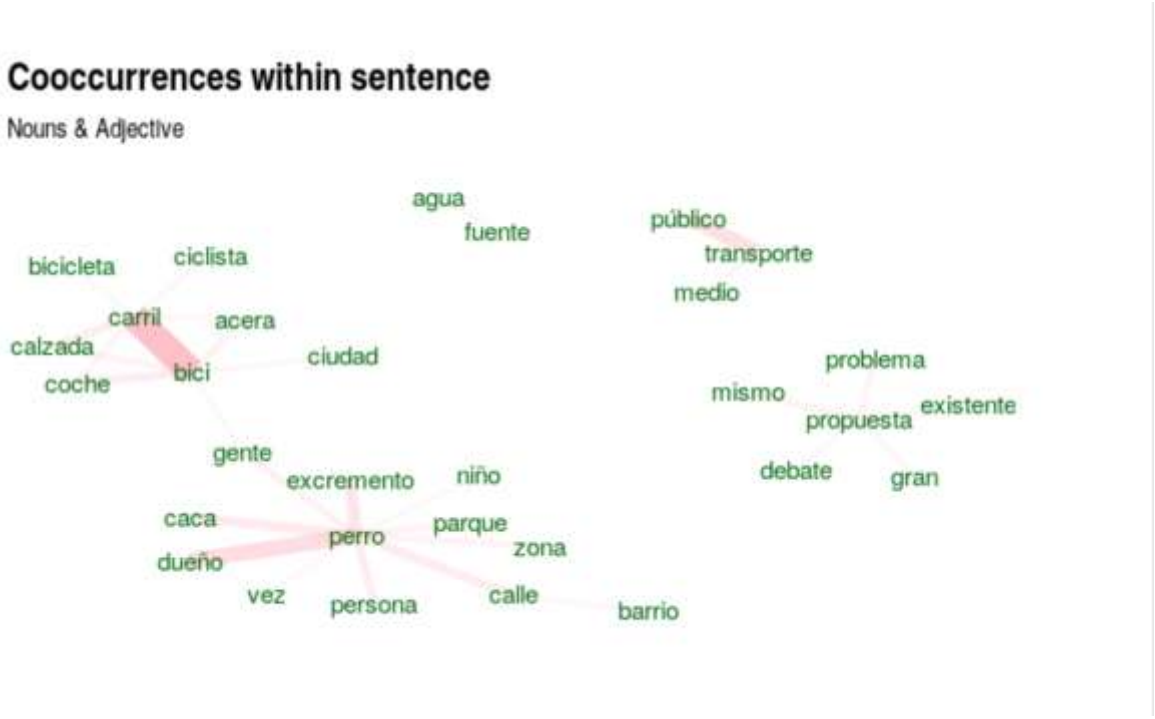

Figura 18. *Mapa de conjuntos de términos relacionados clave.*

En la figura anterior vemos cómo se generan los clusters de conversación alrededor del carril bici, la presencia de perros en la ciudad, el transporte público o incluso las fuentes de agua. Con este análisis, aseguramos que el verdadero tema de interés es el carril bici y no las bicis o los carriles para coches de forma aislada.

Finalmente, podemos representar en forma de **nube de palabras** los temas más importantes, estando seguros de que estamos representando los verdaderos intereses de los ciudadanos en cada momento. Así, almacenando un histórico de estas nubes de palabras podríamos analizar cómo evolucionan los intereses de la ciudadanía a lo largo del tiempo o en función de los diferentes períodos políticos.

Figura 19. *Nube de expresiones más comunes y resultado de los temas más tratados en el conjunto de datos de comentarios ciudadanos.*

Por último, un sencillo análisis de sentimiento nos proporciona una potente herramienta para determinar aquellos debates que tienen una polarización más positiva y negativa. A partir de aquí podríamos querer enfocarnos más en aquellos debates con una polaridad más negativa para intentar anticipar problemas.

```
sentiments <- txt_sentiment(x,
                        term = "lemma",
                        polarity_terms = data.frame(term = c("molesto", "gusta",
"doloroso", "bueno", "mejor", "buena", "dificil", "facil"),
                                                    polarity = c(-1, 1, -1, 1, 1, 1,-1,
1)),
                        polarity_negators = c("no", "tampoco", "nada"),
                        polarity_amplifiers = c("bonito", "mucho", "de verdad", "lo
que"),
                        polarity_deamplifiers = c("ligeramente", "algo"),
                        constrain = TRUE, n_before = 4,
                        n_after = 2, amplifier_weight = .8)

sentiments <- sentiments$data
```

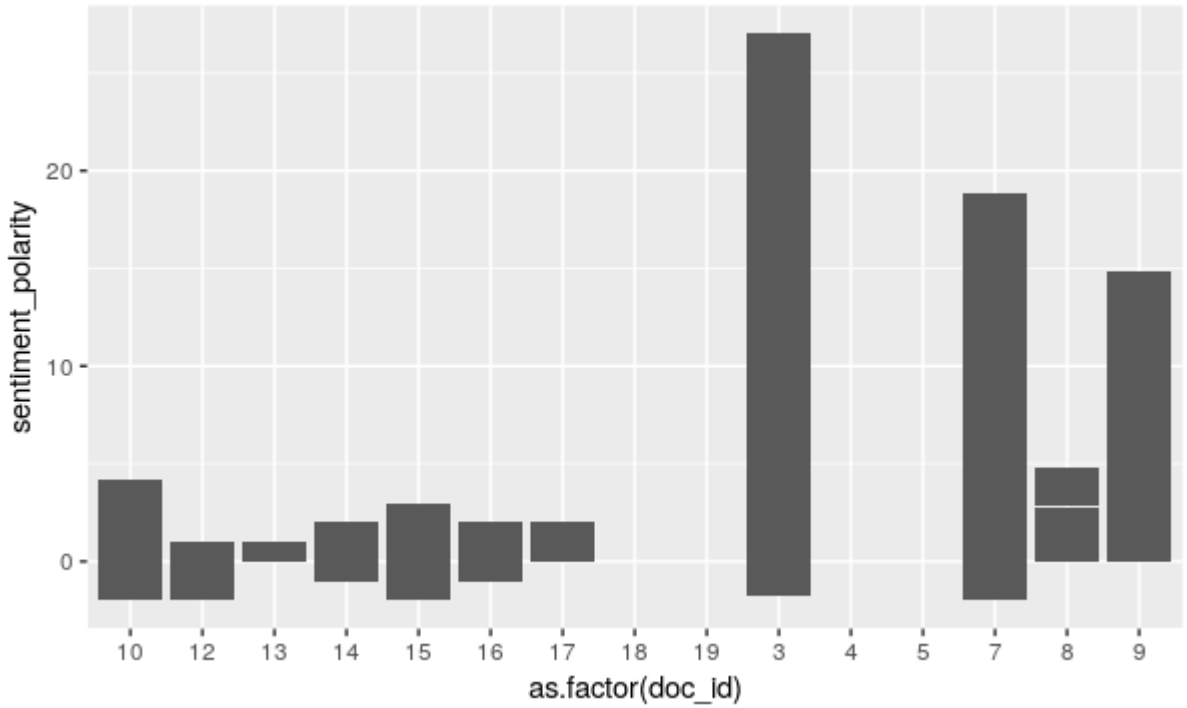

Figura 20. *Análisis de sentimiento y polaridad de los debates en base al número de comentarios positivos y negativos.*

Vemos que **el debate número 3 tiene comentarios fundamentalmente positivos**. Sin embargo, **los debates 12 y 15 tienen múltiples comentarios negativos.** Si recuperamos la descripción de estos debates del fichero original debates.csv vemos que el debate número 3 fue el debate inicial con el que se inauguró el espacio de debates. De ahí que los comentarios sean fundamentalmente positivos, a favor de disponer de este espacio de debate. Los debates 12 (Publicidad sexual en coches) y 15 (Limpieza de las calles) son temas más controvertidos, de ahí que se note el aumento de comentarios negativos.