# TECNOLOGÍAS EMERGENTES Y DATOS

## Procesamiento del lenguaje

Abril 2020

GOBIERNO DE ESPAÑA
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL
SECRETARÍA DE ESTADO DE DIGITALIZACIÓN E INTELIGENCIA ARTIFICIAL
red.es
Iniciativa aporta
datos.gob.es
reutiliza la información pública

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

Este documento ha sido elaborado en el marco de la Iniciativa Aporta (datos.gob.es), desarrollada por el Ministerio de Asuntos Económicos y Transformación Digital a través de la Entidad Pública Empresarial Red.es.

# NOTA EXPLICATIVA

No sabría decir cuánto conocimiento de la humanidad está disponible en texto escrito en multitud de idiomas, pero estoy seguro de que mucho. Esta es la razón por la que los humanos modernos llevamos tanto tiempo intentando escribir programas de software que entiendan el lenguaje humano.

No podemos decir que hayamos enseñado a las máquinas a entender nuestro lenguaje definitivamente, pero si hemos hecho grandes progresos en este campo. En la actualidad, somos capaces de descargar a las personas de tareas tediosas y repetitivas utilizando los últimos avances en **Procesamiento del Lenguaje Natural** (NLP, en sus siglas en inglés). Este campo -a medio camino entre la lingüística y las ciencias de la computación- combinado con los últimos avances en Inteligencia Artificial (AI), está revolucionando algunas áreas de la sociedad y la economía cómo el marketing, la comunicación o la investigación.

En esta nueva entrega de la colección ***Awareness, Inspire y Action*** desgranamos algunos de los misterios del Procesamiento del Lenguaje Natural y tratamos de inspirar al lector con los casos de aplicación más reconocibles y cotidianos. Finalmente, en *Action*, realizamos el desarrollo de un caso de análisis de comentarios ciudadanos extraídos de una plataforma digital de datos abiertos y públicos, desde una perspectiva más técnica y formal.

# 1. INTRODUCCIÓN

*"Oye, Siri"*, *"Ok, Google"*, o *"Hey Mercedes"* son expresiones cotidianas para una parte de la población y, sin embargo, muy pocas personas son conscientes de los fundamentos científicos y tecnológicos que residen debajo de estos comandos tan sencillos. Cuando la complejidad tecnológica es transparente al usuario final es síntoma de que la tecnología está en su punto de madurez óptimo. Así que, **si alguna vez has intentado dialogar con un asistente virtual, eso es fundamentalmente, Procesamiento del Lenguaje Natural**, en adelante NLP (del inglés, *Natural Language Processing*), una de las ramas de las Tecnologías del Lenguaje.

Las tecnologías para el Procesamiento del Lenguaje Natural están mucho más cerca de nuestro día a día de lo que podemos pensar inicialmente. Aplicaciones como la **traducción automática de textos**, el **análisis de sentimiento en redes sociales**, las **búsquedas que realizamos en Internet**, la **generación de los resúmenes** del tiempo o de eventos deportivos, o las peticiones simples que hacemos a nuestro **altavoz *inteligente***, tienen una fuerte componente tecnológica de Procesamiento del Lenguaje Natural.

Sobre el **impacto del NLP en el mundo**, un informe de 2017 de Tractica estima que la oportunidad total de mercado de software, hardware y servicios de NLP será de más de **22.000 millones de dólares (americanos) para 2025**. El documento también pronostica que las soluciones de software de NLP que aprovechan las **capacidades de la Inteligencia Artificial** verán un crecimiento del mercado de **5.400 millones de dólares para 2025**.

En este contexto, desde la Iniciativa Aporta hemos elaborado el informe "Tecnologías emergentes y datos abiertos: Procesamiento del Lenguaje Natural", donde **veremos en profundidad algunos de los casos de uso más relevantes del NLP en la**

**actualidad**. Además, entenderemos la historia del Procesamiento del Lenguaje Natural, sus comienzos y los hitos más importantes de su evolución hasta la actualidad. En la última parte de este informe **analizaremos, de forma práctica, un caso de NLP** (el análisis de comentarios ciudadanos) **aplicado a un conjunto de datos abiertos** con resultados muy interesantes. ¡Te esperamos en las próximas secciones!

# 2. METODOLOGÍA

Este informe se enmarca dentro de una colección más amplia de recursos sobre tecnologías emergentes y datos abiertos, cuyo objetivo es **introducir en la materia al lector mediante el empleo de casos de uso prácticos, sencillos y reconocibles**. Al mismo tiempo, se pretende facilitar **una guía de aprendizaje práctica** para aquellos lectores con conocimientos más avanzados, que, mediante el desarrollo de un caso práctico, puedan experimentar de forma autodidacta con herramientas reales para el análisis y explotación de datos abiertos.

Para conseguir este doble objetivo, el informe se estructura en tres partes bien diferenciadas: *Awareness*, *Inspire* y *Action* (Figura 1), que pueden ser abordadas de forma independiente en cualquier momento y sin necesidad de haber realizado una lectura previa de las otras secciones.


***Figura 1.*** *Metodología de la colección Awareness, Inspire, Action.*

La primera sección, ***Awareness,*** sirve de introducción al tema en cuestión (en este informe, el Procesamiento del Lenguaje Natural). Esta sección está indicada para aquellas personas que se inician en el tema por primera vez y tratan de abordar la temática de forma sencilla, clara y sin el uso de tecnicismos que dificulten la lectura.

La segunda sección, ***Inspire,*** pretende servir de inspiración a aquellas personas que se han iniciado en la materia y que se preguntan cómo les afecta a ellas en su vida diaria o en su trabajo el tema que se aborda. La forma de identificarse con una tecnología, un campo de la ciencia o cualquier otra materia es verse reflejado en ella. De esta forma, la sección *Inspire,* contiene ejemplos y casos de aplicación de una cierta tecnología en situaciones, más o menos, cotidianas que favorece que los lectores se identifiquen y comiencen a pensar en dicha tecnología como algo que también les afecta.

Por último, la sección ***Action*** selecciona alguno de los casos de usos explicados en la sección *Inspire* y lo desarrolla de forma práctica, utilizando para ello, datos y herramientas tecnológicas reales. El ejemplo, desarrollado en Action, se pone a disposición de las personas usuarias de esta guía en forma de código y datos abiertos (Anexo I) para que puedan experimentar y desarrollar con sus propios medios el caso de uso que se aborda en la sección *Action*.

# 3. AWARENESS

## 3.1 ¿Qué son las Tecnologías del Lenguaje?

Las **tecnologías digitales del lenguaje** son aquellas capacidades, herramientas informáticas y algoritmos que hacen posible que las **máquinas puedan entender y generar expresiones en lenguaje humano** (escrito y hablado) en **múltiples idiomas**. El conjunto de tecnologías digitales del lenguaje ocupa un lugar preferente y de actualidad en los principales *hubs* y espacios de innovación de las empresas e instituciones académicas. También es habitual que el apoyo al desarrollo de estas tecnologías se materialice en forma de planes y programas impulsados por las instituciones estatales, como en el caso de España con el **Plan de Impulso a las Tecnologías del Lenguaje.**

El siguiente cuadro muestra el marco teórico establecido por el Ministerio de Asuntos Económicos y Transformación Digital a través del citado Plan, donde se habla de *Procesamiento del Lenguaje Natural (PLN[1]) y Traducción Automática (TA).*

[1] En el Plan de impulso a las tecnologías del lenguaje, elaborado por el Ministerio de Asuntos Económicos y Transformación Digital, las siglas utilizadas para referirse al Procesamiento del Lenguaje Natural son PLN. Sin embargo, en este informe utilizaremos su forma en inglés NLP (Natural Language Processing). Esto se debe a que el uso de estas siglas en inglés está mucho más extendido. De esta forma, si un lector interesado en el tema, realiza una búsqueda en Google del término PLN, no encontrará ninguna entrada relevante en la primera página de resultados. De lo contrario, cuando buscamos NLP, rápidamente accedemos a la información general sobre Procesamiento del Lenguaje Natural.

*Las tecnologías de Procesamiento del Lenguaje Natural (PLN) y Traducción Automática (TA) son las tecnologías que hacen posible analizar textos y facilitar su explotación en aplicaciones informáticas de uso muy común en sectores tan dispares como la Sanidad, la Educación o el Turismo.*

*Por ejemplo, la detección de entidades nombradas (nombres propios de personas o empresas, marcas de productos o topónimos), filtrado y clasificación de documentos, creación de resúmenes automáticos, extracción de información, análisis de sentimientos, minería de opinión, seguimiento y monitorización de la reputación en los medios sociales, corrección ortográfica y gramatical, búsqueda inteligente y optimizada, sistemas de respuesta automática a preguntas y asistentes personales, la traducción automática de textos, etc.*

*Todas estas aplicaciones se pueden resumir como la explotación de información no estructurada que mejora la comprensión de textos en corpora documentales.*

La visión tan multidisciplinar del conjunto de tecnologías del lenguaje se representa bien en la siguiente figura:


***Figura 2.*** *¿Cuál es el lugar del Procesamiento del Lenguaje Natural como campo de la ciencia y la tecnología? Fuente: https://www.plantl.gob.es/tecnologias-lenguaje/catalogo-TL/Paginas/clasificacion-TL.aspx.*

De acuerdo con este mismo marco teórico, las principales aplicaciones de este conjunto de tecnologías son:

- Optimización de procesos industriales de gestión lingüística de documentación: traducción de documentos y herramientas de autor (correctores, generación de documentos, etc.)
- Comunicación y asistencia personal (asistentes virtuales, comunicación persona-máquina para coches, atención al cliente e interacción con robots; buscadores inteligentes y respuesta automática de preguntas).
- Procesamiento inteligente de información y conocimiento (extracción y minería de información de textos y contenidos, clasificación de documentos, resumen automático, etc.).
- Asistencia en el aprendizaje de lenguas.

## El Procesamiento del Lenguaje Natural

**En particular,** este informe se enfoca fundamentalmente sobre el campo del **Procesamiento del Lenguaje Natural**, que tiene **múltiples aplicaciones en nuestra vida cotidiana**.

De una forma sencilla podemos decir que el **Procesamiento del Lenguaje Natural (NLP)** es hacer que las máquinas (los ordenadores) **entiendan el lenguaje humano**: hablado o en forma de texto. Más formalmente, como hemos introducido anteriormente, el Procesamiento del Lenguaje Natural (NLP) es un campo híbrido entre la informática y la lingüística, que utiliza diferentes técnicas, algunas de ellas basadas en Inteligencia Artificial, para interpretar el lenguaje humano.

## Procesamiento de Lenguaje Natural (NLP)

Formalmente, el NLP es un campo interdisciplinar que trata de hacer que las máquinas, mediante programas de software, sean capaces de leer, entender y derivar el significado del lenguaje humano escrito. Las aplicaciones del NLP es la vida cotidiana son múltiples. Algunos ejemplos populares son:

- **Autocompletar y predicción de texto:** en motores de búsqueda (por ejemplo: Google, Bing) y más recientemente en los clientes de correo electrónico.
- **Revisión ortográfica:** en casi todas partes, en el navegador, en el procesamiento de textos (por ejemplo: Microsoft Office u Open Office) en las aplicaciones de mensajería instantánea.
- **Análisis de revisiones y comentarios**: Analizar automáticamente las opiniones (webs de recomendación sobre productos, restaurante, viajes, etc.) de los clientes es una de las mayores aplicaciones de NLP.

Existe bastante ambigüedad en la bibliografía existente sobre la definición y el uso del término NLP (Procesamiento del Lenguaje Natural, en español).

- Multitud de referencias[2] utilizan NLP para referirse, de forma general, al conjunto de tecnologías que hacen posible que seamos capaces de comunicarnos con una máquina independientemente del idioma y el canal que utilicemos para ello. Por lo tanto, este uso del término NLP agrega todas aquellas tecnologías de traducción automática (TA), asistentes conversacionales y sistemas de conversión de lenguaje hablado a texto y viceversa.
- En otras ocasiones, sin embargo, encontramos usos más concretos del término NLP, refiriéndose estrictamente a aquellas tareas que tienen que ver con el análisis de las oraciones escritas para su simplificación y posterior clasificación.

Sea como fuere, el propósito de este informe es introducir al lector en el campo de las tecnologías digitales que, de una forma u otra, sin entrar en demasiados tecnicismos, son responsables de que las máquinas sean capaces de entender el lenguaje humano para multitud de aplicaciones.

Precisamente, uno de los objetivos de este informe es plantear de forma clara y cercana los múltiples casos de uso que se apoyan en este conjunto de tecnologías. En este sentido, a lo largo del documento se utiliza el concepto de NLP desde una perspectiva amplia para atribuir a un solo término las múltiples aplicaciones de este grupo de tecnologías.

[2] Otros ejemplos:

- https://www.datacentric.es/blog/business/procesamiento-lenguaje-natural-revolucion-futuro/
- https://www.sas.com/es_ar/insights/analytics/what-is-natural-language-processing-nlp.html

## 3.2 La importancia de los datos abiertos en el Procesamiento del Lenguaje Natural

Los **algoritmos de NLP son grandes consumidores de datos** que sirven de entrada para el **entrenamiento de los modelos de Inteligencia Artificial** que hacen posible el entendimiento del lenguaje humano por parte de las máquinas. Además, la particularidad del procesamiento de lenguaje natural frente a otros campos de la ciencia de datos es su **variabilidad idiomática**. Esto es, los modelos han de ser entrenados con conjuntos de datos en cada idioma para obtener resultados óptimos. Veamos un ejemplo para hacernos una idea de la magnitud de los datos necesarios.

Uno de los últimos algoritmos de NLP publicados en 2019, GPT-2, ha sido entrenado con **40GB de textos disponibles en Internet**. Por comparación, una copia de *El Quijote* de Miguel de Cervantes en formato *pdf* ocupa aproximadamente un 1MB de espacio en disco. De forma ilustrativa, el algoritmo GPT-2 ha sido entrenado con 40.000 obras del tamaño de *El Quijote*. Es evidente, que tal cantidad de texto escrito, **necesita necesariamente del uso de datos abiertos en forma de textos**. Algunos repositorios de datos abiertos están especialmente preparados para albergar textos que sirvan como recursos lingüísticos de calidad para entrenar algoritmos de NLP. En el reciente informe *Estudio sobre datos reutilizables como recursos lingüísticos* (2019) se describe una relación de recursos lingüísticos por temática y organización de origen.

En el momento de escribir este informe, el mundo entero se enfrenta a una de las mayores pandemias de la era moderna. **La crisis del Covid-19** eclipsa cualquier otra noticia de interés. Incluso en esta grave situación de emergencia sanitaria mundial, las **tecnologías de NLP junto con los datos abiertos juegan un papel fundamental** para ayudar a la sociedad en la lucha contra el virus. Así, La Casa Blanca junto con una coalición de grupos de investigación líderes han preparado el conjunto de datos abiertos sobre la investigación del COVID-19 (CORD-19). El conjunto de datos CORD-19 es un recurso de más de 44,000 artículos académicos, incluidos más de 29,000 con texto completo, sobre COVID-19, SARS-CoV-2 y coronavirus relacionados. Este conjunto de **datos de libre acceso** se proporciona a la comunidad de investigación global para aplicar los avances recientes en el **Procesamiento del Lenguaje Natural** y otras técnicas de IA para generar nuevas ideas en apoyo de la lucha continua contra esta enfermedad infecciosa.

## 3.3 ¿Cómo hacemos que las máquinas entiendan el lenguaje humano?

El ser humano ha sido muy hábil con el desarrollo de los ordenadores y la ciencia de la computación moderna. Un ordenador convencional basado en tecnología del silicio es una máquina, que, a pesar de su complejidad, se basa en el simple principio de **codificar y decodificar información digital binaria** basada en ceros y unos. Por lo tanto, parece lógico pensar que, para hacer que una máquina *entienda* nuestro lenguaje, debemos de convertir el texto en códigos binarios. Esto se conoce como **codificación de texto o *text encoding***.

Es importante destacar que la máquina, y en particular, los modelos (algoritmos) de Procesamiento del Lenguaje Natural no *entienden,* en sentido estrictamente humano, el significado de nuestro lenguaje. En realidad, estos modelos lo que hacen es **mapear la estructura estadística del lenguaje escrito**. Habitualmente esta fórmula es suficiente para resolver muchas tareas textuales simples como las que hemos citado en párrafos anteriores.

Veamos un ejemplo sencillo para entender cómo funciona el *text enconding*. Por simplicidad vamos a analizar una frase **a nivel de sus palabras**. Podríamos utilizar caracteres individuales u otras estructuras como conjuntos de varias palabras o expresiones. En NLP, a los conjuntos de palabras que forman una expresión se les conoce como N-gram donde N representa el número de (en este caso) palabras que tiene la expresión.

Utilizando esta aproximación del análisis de las palabras que forman una oración, tomemos la siguiente frase: "El gato se sentó sobre el libro." Veamos qué pasos son necesarios para ejecutar el proceso del *text encoding*.

Lo primero que hacemos es **asignar un índice** a cada palabra de la siguiente forma:

| El | gato | se | sentó | sobre | el | libro. |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |

Una vez hecho esto **generamos un array** (en este caso una lista de vectores) **que representa el índice de cada palabra y su posición de ocurrencia**. En este caso la única palabra que se repite es "el".

El resultado del *encoding* es el siguiente en forma de lista de 7 vectores (array).

```
, , 1

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    1    0    0    0    0    0    0    0    0     0

, , 2

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    0    1    0    0    0    0    0    0    0     0

, , 3

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    0    0    1    0    0    0    0    0    0     0

, , 4

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    0    0    0    1    0    0    0    0    0     0

, , 5

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    0    0    0    0    1    0    0    0    0     0

, , 6

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    0    0    0    0    0    1    0    0    0     0

, , 7

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,]    0    0    0    0    0    0    1    0    0     0
```

La palabra “el” aparece en la posición 1 y 6 de la oración (que se representan en la figura anterior como ,,*1* y ,,6). Se han generado dos vectores distintos, aunque se trata de la misma palabra. Así, la forma de representar esta ocurrencia de palabras en el array de resultados es como se muestra en la siguiente figura.

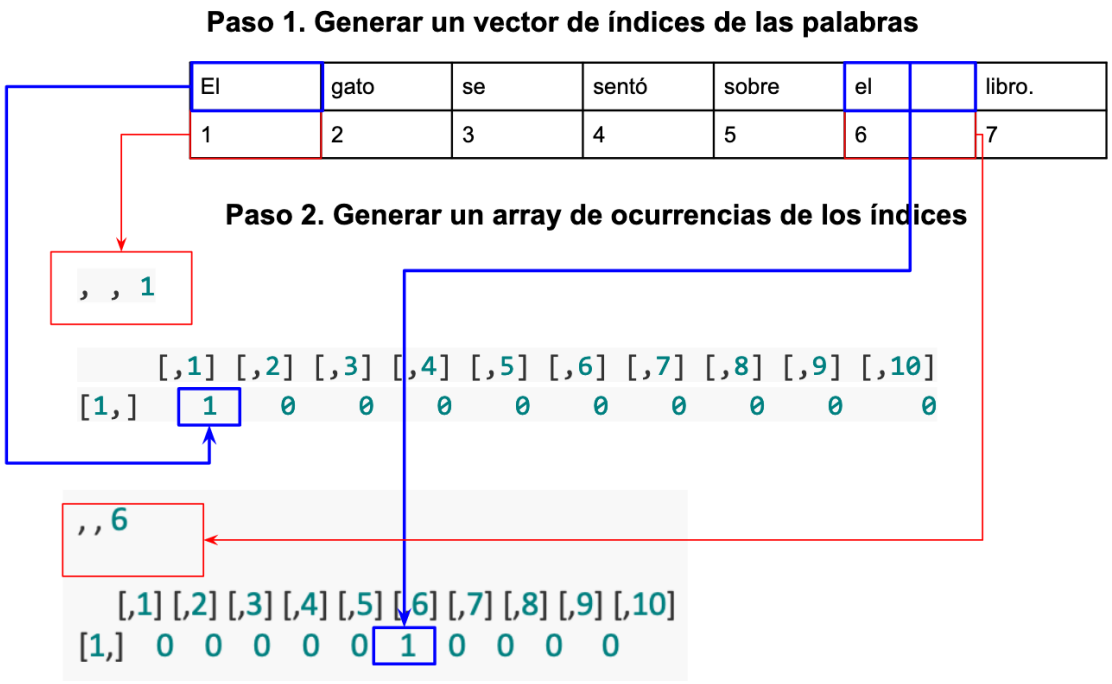



**Figura 3.** *Proceso de word-encoding explicado de forma gráfica.*

La técnica que hemos visto aquí es la más sencilla para codificar texto en representaciones numéricas. Esta técnica se denomina ***one-hot-encoding*** y a pesar de ser ampliamente utilizada presenta las siguientes **limitaciones**:

- Se **utilizan grandes cantidades de memoria** para almacenar información en su mayoría nula o inservible, puesto que la mayoría de las posiciones del array multidimensional que hemos visto anteriormente son cero.
- Se **pierde la información relativa a la similitud de las palabras,** lo que va en detrimento de la capacidad de entender su significado.

Alternativamente a este método, podríamos dar un paso más y generar un vector denso, asignando a cada palabra en la frase un índice arbitrario único. Por ejemplo, ordenaríamos alfabéticamente el vocabulario de la frase: *El gato libro se sentó sobre*. Asignamos un índice numérico a cada palabra del vocabulario ordenado.

Finalmente, construimos la oración colocando el índice en la posición de ocurrencia [1,2,4,5,6,1,3]

La alternativa al método *one-hot-encoding* es el mecanismo conocido como ***word embeddings***. Mientras que los vectores obtenidos a través de la codificación ***one-hot-encoding*** son **binarios**, **dispersos** (la mayoría de los valores son cero) y de muy **alta dimensión** (como vimos en el ejemplo, para una sencilla frase se generaban 7 vectores, uno por palabra), el resultado del ***word embeddings*** son vectores de más **baja dimensión** (vectores densos), a diferencia de los vectores dispersos.

Para grandes vocabularios (por ejemplo, 20.000 palabras), el método *one-hot-encoding* generaría arrays de dimensión 20.000 mientras que el método *word embeddings* generaría arrays de 256 o 512 dimensiones. Por lo tanto, una de las principales ventajas de utilizar *word embeddings* es su capacidad **para comprimir la misma información en muchas menos dimensiones**.

Finalmente, otra gran ventaja del método *word embeddings* es que los vectores densos se aprenden (se generan) de los datos de entrada, mientras que en el *one-hot-encoding* la asignación del índice es arbitraria. Es decir, es posible generar un ***espacio* de *word embeddings* determinado para un conjunto de textos de entrada, un idioma determinado y un objetivo concreto**. Es decir, tienen en cuenta las relaciones entre las palabras. Por ejemplo, París, Grenoble y Francia, tienen similitud en el contexto de países y ciudades, y por lo tanto los números que representan estas palabras serán similares entre sí. Otro ejemplo, un espacio *word embeddings* concreto es aquel que se ha generado a partir de una base de datos de críticas de películas de cine en inglés. El objetivo de este espacio es servir como base para analizar qué películas han gustado más y cuáles menos. Una vez generado este espacio, puede ser utilizado por cualquier

aplicación similar a la anterior. De esta forma, sería como tener un modelo de Inteligencia Artificial pre-entrenado al que solo le tenemos que suministrar los nuevos datos de entrada.

En resumen, las diferencias entre el *one-hot- encoding y el word embeddings* son:

| ONE-HOT-ENCODING | WORD EMBEDDINGS |
|---|---|
| **Binario:** vector formado por 0 y 1. | **Continuo:** vector formado por números reales. |
| **Disperso:** la mayoría de los valores son cero. | **Denso:** los valores son números reales. |
| **Alta dimensión:** se generan un gran número de vectores. | **Baja dimensión:** permite comprimir la misma información en menos vectores. |
| **Codificación impuesta:** los índices se establecen manualmente de manera arbitaria. | **Codificación aprendida:** los valores de los vectores se aprenden de los datos. |

**Figura 4.** *Comparativa entre el método one-hot-encoding y el word embeddings.*

La complejidad técnica que subyace debajo del Procesamiento del Lenguaje Natural hace que sea imposible cubrir con mayor nivel de detalle los procesos de generación de espacios de *word embeddings*. Si bien es cierto que la introducción incluida en párrafos anteriores será muy valiosa para seguir de forma fluida el ejemplo completo de la sección *Action*, incorporar más complejidad técnica a esta informe queda fuera de su alcance. Sin embargo, el lector más atrevido puede consultar la sección *Próxima parada* donde encontrará multitud de enlaces a referencias que extienden con creces el contenido técnico de este informe.

## 3.4 Un poco de historia

La historia del Procesamiento del Lenguaje Natural abarca el periodo que va desde el fin de la Segunda Guerra Mundial hasta nuestros días. Por tanto, cuenta ya **con 75 años de largo y arduo** recorrido.

Alan Turing, conocido como uno de los padres de la Inteligencia Artificial y de los antepasados de los ordenadores, publicó en 1950 un artículo titulado "Computing Machinery and Intelligence", que puede considerarse el texto que inaugura la historia del NLP. Merece la pena citar el comienzo del artículo:

> *I PROPOSE to consider the question: "'Can machines think?"*
>
> *This should begin with definitions of the meaning of the terms 'machine' and 'think'.*
>
> ***TRADUCCIÓN:*** Propongo considerar la pregunta: "¿Pueden las máquinas pensar?" Esto debería comenzar con definiciones del significado de los términos "máquina" y "pensar".

No deja de ser sorprendente que alguien comenzara así un artículo sobre la inteligencia de las máquinas hace 70 años, teniendo en cuenta que lo más parecido a un ordenador era un engendro mecánico del tamaño de una habitación.

**Figura 5**. *De Karl Baron from Lund, Sweden - Vacuum tube computer: Uploaded by shoulder-synth, CC BY 2.0,* https://commons.wikimedia.org/w/index.php?curid=8828095

Hasta la década de 1980, la mayoría de los sistemas de procesamiento de lenguaje natural se basaban en conjuntos complejos de reglas pre-definidas, coincidiendo a su vez con el auge de los denominados sistemas expertos[3].

A partir de finales de los **años 80**, se produce la primera revolución en el campo del Procesamiento del Lenguaje Natural. Gracias al **aumento en la capacidad de cálculo**

[3] Los sistemas expertos son programas informáticos que contienen reglas lógicas que codifican y parametrizan el funcionamiento de sistemas sencillos. Por ejemplo, un programa informático que codifica las reglas del juego de ajedrez pertenece al tipo de programas que conocemos como sistemas experto.

de los ordenadores siguiendo la Ley de Moore, comienzan a introducirse **estrategias basadas en la estadística avanzada** (primeros algoritmos de machine learning) para el procesamiento del lenguaje. Algunos de estos antiguos algoritmos de machine learning, como los árboles de decisión, producían sistemas de reglas estrictas similares a las diseñadas manualmente en la década anterior. Con la progresiva **democratización de los ordenadores personales**, se generaron **más y más datos digitales** de entrada para entrenar a estos algoritmos, mejorando de forma continua su precisión en tareas como la clasificación de textos, dando como resultado los filtros anti-spam, por ejemplo.

El siguiente hito importante en el campo del procesamiento del lenguaje se produce en el año **2013**, cuando el grupo de investigación de Google dirigido por Tomas Mikolov inventan el **algoritmo Word2vec**. A partir de la existencia de algoritmos como word2vec y otros posteriores como Glove o FastText que pueden ser pre-entrenados con grandes volúmenes de datos, el campo del NLP sufre una gran democratización, permitiendo a los desarrolladores de software crear aplicaciones que utilizan el Procesamiento del Lenguaje Natural como entrada o salida de la funcionalidad de dicha aplicación.

En resumen, la historia del Procesamiento del Lenguaje Natural es muy amplia, como resume la siguiente imagen:

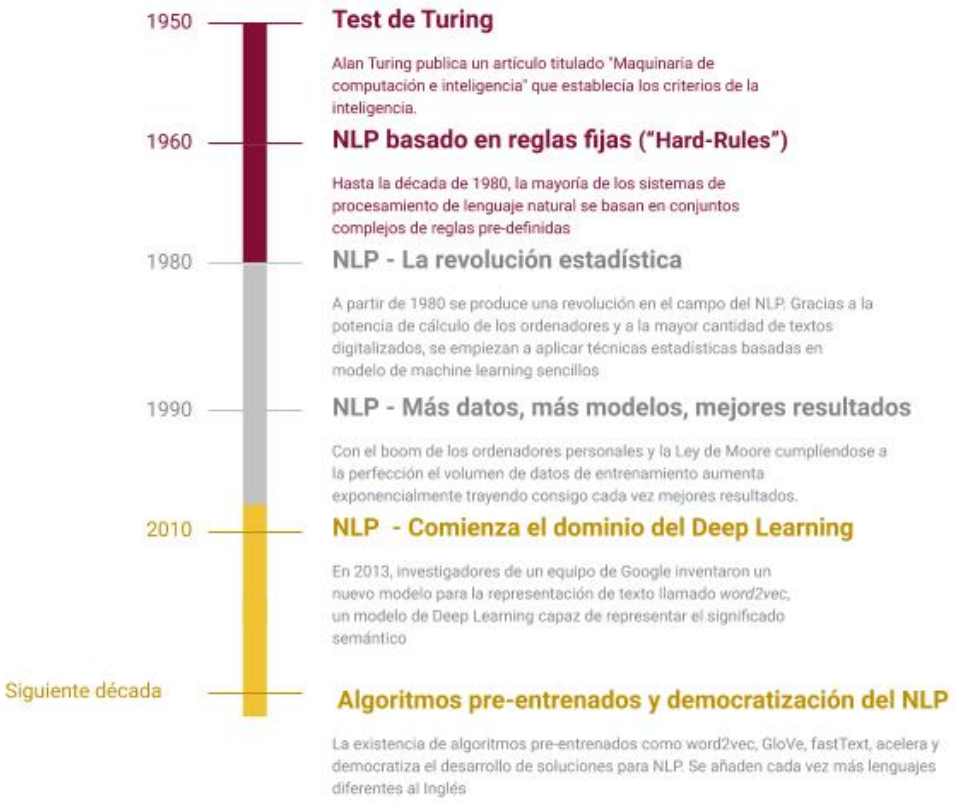



**Figura 6.** *Línea temporal que destaca los hitos más importantes en el desarrollo del NLP desde sus inicios hasta nuestros días.*

## 3.5 Impacto

Desde un punto de vista de impacto tangible, el NLP ha realizado grandes avances en los últimos años, impulsado por la ola de digitalización que vivimos y navegando por el infinito océano de Internet. A continuación, revisamos los principales hitos del NLP divididos por su grado de madurez:

**Tareas casi resueltas completamente por NLP**

**Detección de spam.** ¿Desde cuándo no revisas el Spam en tu cuenta de correo electrónico? No hace muchos años que había que comprobar varias veces al día que un correo electrónico importante no había terminado en la bandeja del Spam de tu cliente o servidor de correo. En la actualidad, la probabilidad de que un algoritmo de detección de spam produzca un falso positivo es realmente baja.

**Detección de partes de las oraciones** (POS, del inglés *Part of Speech*). Son algoritmos que, dada una oración, tratan de determinar qué tipo de palabras se encuentran en ella, por ejemplo, cuáles son nombres, verbos, adjetivos, etc.

**Detección y reconocimiento de entidades** (NER, del inglés *Name Entity Recognition*). Son algoritmos que, dada una oración, tratan de determinar a qué tipo de entidades corresponden los nombres que se encuentran. Por ejemplo, nombres de personas, ubicaciones, organizaciones.

**Tareas que demuestran un rápido y satisfactorio avance en NLP**

**Análisis de sentimientos.** Dada una oración, el algoritmo trata de determinar su polaridad (por ejemplo: positiva, negativa, neutral) o emoción (por ejemplo: feliz, triste, sorprendida, enojada). Esta tarea tiene

una gran importancia en el análisis de opinión que, lógicamente, es de crucial importancia para empresas de productos, servicios, medios de comunicación, etc.

**Detección de referencias cruzadas**. Para hacer que una máquina entienda el lenguaje humano es necesario detectar qué palabras hacen referencias unas a otras. Por ejemplo, en una oración en la que hay un nombre propio y más adelante se usa un pronombre para referirse al nombre anterior, es necesario detectar que ambos, nombre propio y pronombre, hacen referencia a lo mismo dentro del significado de la oración.

**Desambiguación del sentido de las palabras** (WSD, del inglés *Word Sensing Desambiguation*). En el lenguaje humano, muchas palabras tienen más de un significado. Para entender el significado de una oración en particular, es necesario seleccionar el significado que más sentido tenga en el contexto de dicha oración.

**Tareas de NLP cuyo grado de madurez es todavía limitado**

**Asistentes de diálogo y chat-bots.** Aunque su evolución ha sido notoria en los últimos años, todavía son tecnologías de uso muy restringido y limitado a dominios muy específicos (medicina, asistente de call-center, acciones rutinarias con el smartphone, etc.).

**Asistentes de pregunta-respuesta.** Su capacidad de entender el sentido de la pregunta especialmente en lenguaje hablado es bajo y las acciones de call-back (es decir, aquellas acciones que el asistente tiene que realizar cuando no encuentra lo esperado por la persona usuaria o no ha entendido el sentido de la pregunta) son muy rudimentarias.

**Generación de resúmenes.** La generación de resúmenes de texto pertenece a un subdominio del NLP conocido como generación de lenguaje natural (NLG). El grado de desarrollo del NLG es bajo fuera de dominios específicos y entornos con condiciones muy controladas (como, por ejemplo, resúmenes de eventos deportivos basados en estadísticas, o resúmenes meteorológicos).

**NLP para idiomas de bajos recursos.** Se estima que en el mundo hay unos 7.000 idiomas hablados y, sin embargo, la mayoría de estos idiomas son residuales e incapaces de generar suficiente material escrito para poder entrenar los algoritmos de procesamiento. Las últimas investigaciones en NLP ponen el foco en estos idiomas con nuevas técnicas que permiten mitigar el efecto de disponer de recursos escasos para el procesamiento.

Ahora que ya conocemos los principios básicos del Procesamiento del Lenguaje Natural (incluso nos hemos asomado ligeramente a sus bases técnicas), su historia y sus principales aplicaciones, es el momento de activar nuestra inspiración visitando juntos algunos casos de uso de mucha actualidad que se sustentan sobre las bases del NLP.

# 4. INSPIRE

En esta sección veremos con más detalle algunos de los **casos de uso particulares** del **Procesamiento del Lenguaje Natural y sus aplicaciones prácticas**. Varios casos de uso descritos en el primer informe de la serie ***Awareness, Inspire y Action*** , titulado Tecnologías emergentes y datos abiertos: Inteligencia Artificial, tienen relación directa con las tecnologías de Procesamiento del Lenguaje Natural. Sin duda, la evolución presente y futura del NLP, depende en gran medida de los últimos avances en Inteligencia Artificial.

Algunos ejemplos de casos de uso ya han sido introducidos en la sección de *Awareness*. Ahora es el momento de profundizar en la predicción de texto, la clasificación de textos y la detección de fake news, tres interesantes ejemplos que nos dan una visión bastante amplia de las posibilidades de estas tecnologías.

## 4.1 Predicción de texto

Quizás uno de los avances más palpables en el campo del Procesamiento del Lenguaje Natural sea la **predicción de texto**. En los últimos meses hemos visto cómo las últimas actualizaciones de los principales clientes de correo electrónico y motores de búsqueda, traían consigo **una funcionalidad sorprendente**. Cuando escribimos las primeras letras (incluso antes) de un email o una búsqueda web, el motor de procesamiento de lenguaje natural, utiliza un modelo entrenado que predice las palabras que vienen a continuación en un ranking de probabilidad. Es sorprendente el buen funcionamiento de esta funcionalidad y nos confirma que los humanos funcionamos en base a patrones de comportamiento regular. Cada persona tiene una

huella sobre cómo se expresa, las oraciones que utiliza con más frecuencia, los comienzos y finales de las conversaciones, etc.

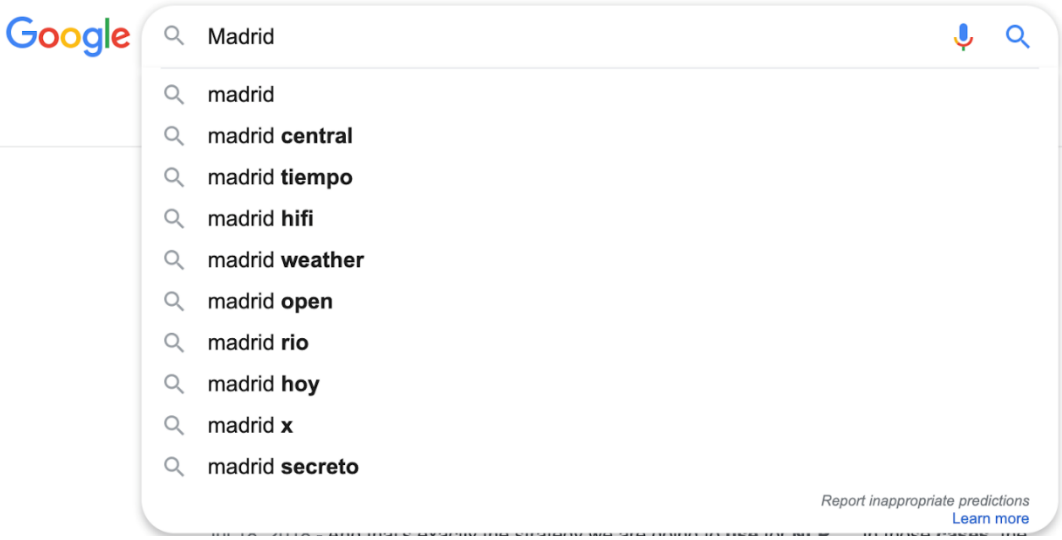



**Figura 7.** *Motor de palabras relacionadas de Google.*

En la sección *Action* veremos cómo el algoritmo RAKE (Rapid Automatic Keyword Extraction) permite extraer **conjuntos de términos** que aparecen juntos o correlativos con mayor probabilidad.

## 4.2 Clasificación de textos

La clasificación automática de textos es vital para muchas empresas y aplicaciones en la actualidad. La **detección de correos electrónicos fraudulentos** es un caso particular de la clasificación de textos. También lo son las **valoraciones de comentarios online**, por ejemplo, para valorar productos en base a los comentarios de las personas usuarias.

Los clasificadores de texto son tan útiles en la actualidad por varias razones:

1) **Son rápidos y sencillos.** Los algoritmos de clasificación de texto lineales son tan simples (en comparación con los modelos de Inteligencia Artificial más

complejos como, por ejemplo, redes neuronales recurrentes) que se pueden entrenar rápidamente en un ordenador corriente. Además, la diferencia de precisión entre este tipo de algoritmos y otros mucho más complejos es casi nula, haciendo que no merezca la pena el sobreesfuerzo en el entrenamiento.

2) **Son casi independientes del idioma.** Para un algoritmo de clasificación no importa en qué idioma está el texto siempre que pueda separarse en palabras y medir los efectos de esas palabras.
3) **Su precisión es muy alta,** casi comparable con los seres humanos. El número de falsos positivos en detección de correo fraudulento o en identificación del género de una película o libro es muy bajo, casi despreciable.


**Figura 8.** *Flujo simplificado de un motor de detección y clasificación de correo fraudulento.*

La forma más sencilla de crear tu propio clasificador de textos es utilizar la herramienta de código abierto fastText, creada originalmente por Facebook.

## 4.3 Fake News

No todas las aplicaciones del Procesamiento del Lenguaje Natural tienen el objetivo de servir a un buen propósito. Recientemente estamos asistiendo a la explosión de las **fakes news o noticias falsas**: información falsa creada de forma deliberada y publicada a través de las redes sociales o diarios electrónicos con el objetivo de polarizar la opinión pública en un determinado sentido u orientación.

Es ya un hecho que las fake news se han utilizado de forma masiva en campañas de desprestigio político. Además de las fake news, recientemente, han salido a la luz los conocidos **deep fakes o videos falsos** que aplican la misma fórmula para modificar videos en vez de textos. En estos vídeos, se sustituye el rostro y la voz del verdadero protagonista por otra persona, habitualmente famosa, de la que existe gran cantidad de fotografías y audios en Internet. Si bien la calidad de los deep fakes está lejos de ser perfecta, en el caso de las fake news escritas es prácticamente imposible distinguirlas de las noticias reales salvo que se sepa específicamente que el contenido es falso.

En febrero de 2019, OpenAI anunció una nueva arquitectura de Procesamiento del Lenguaje Natural llamada **GPT-2**. GPT-2 es capaz de **generar fragmentos de texto absolutamente realistas** a partir de tan solo un par de palabras cómo comienzo de la frase.

Los resultados de GPT-2 son tan sorprendentes como inquietantes. Sus propios creadores dicen en su página web *Politicians may want to consider introducing penalties for the misuse of such systems, as some have proposed for deep fakes. (*Los políticos pueden considerar introducir sanciones por el mal uso de tales sistemas, como algunos han propuesto para los deep fakes).

Veamos un ejemplo de lo que GPT-2 es capaz de hacer. Adam Geitgey, en su blog de Medium, nos enseña un ejemplo muy ilustrativo.

Si utilizamos GPT-2 con un fragmento de texto inicial como **Abraham Lincoln** se genera una oración que se ajusta a este personaje histórico:

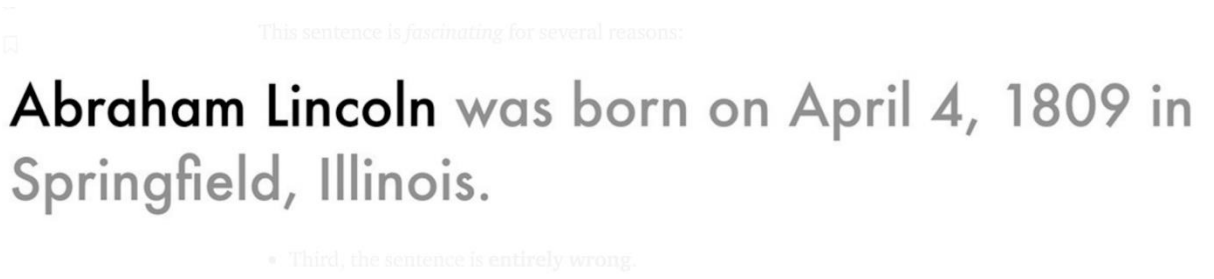



**Figura 09.** *Frase generada automáticamente por el algoritmo GPT-2.*

La frase es fascinante por varias razones:

- **Primero**, muestra que el modelo ha *entendido* que Abraham Lincoln era una persona (figura histórica) en Estados Unidos que nació en 1809.
- **Segundo**, la oración está perfectamente escrita e indistinguible de algo escrito por un ser humano.
- **Tercero**, la afirmación es completamente falsa.

Abraham Lincoln nació en Estados Unidos el 12 de abril de 1809 en Kentucky y no el 4 de abril en Illinois. Pero, sinceramente, yo no lo sabía, lo he tenido que consultar en Wikipedia y en ningún momento me ha parecido sospechoso ni alarmante la afirmación.

A pesar de los usos maliciosos que se puedan derivar del uso de esta tecnología, los algoritmos que subyacen son siempre un arma de doble filo. De la misma forma que generan noticias falsas con un asombroso realismo, se pueden reorientar a la detección de noticias falsas y la lucha contra las emergentes fake news.

# 5. 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.

## 5.1 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.

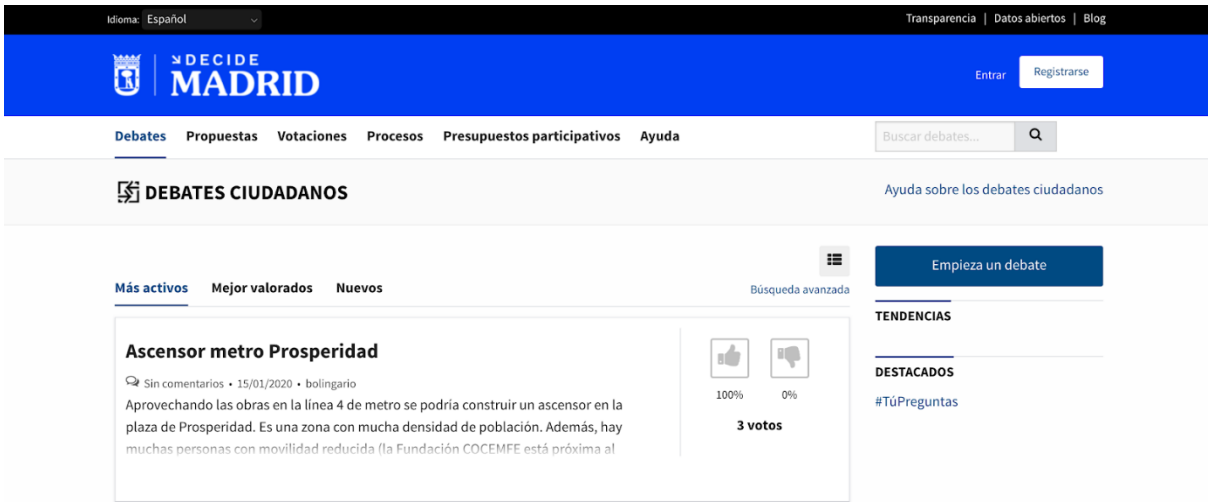



**Figura 10.** *Plataforma digital Madrid decide*. http://decide.madrid.es.

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_vot |
|---|---|---|---|---|---|
| 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.*

## 5.2 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 | 3 | 1 | 1 | Ilusionante sin duda. | 1 | Ilusionante | Ilusionante | ADJ | NA | NA |
| 2 | 3 | 1 | 1 | Ilusionante sin duda. | 2 | sin | sin | ADP | NA | NA |
| 3 | 3 | 1 | 1 | Ilusionante sin duda. | 3 | duda | duda | NOUN | NA | Gender=Fem\|Number=Sing |
| 4 | 3 | 1 | 1 | Ilusionante sin duda. | 4 | . | . | PUNCT | NA | NA |
| 5 | 3 | 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 | 3 | 2 | 1 | Queda por ver cómo se consigue articular y qué tal se... | 2 | por | por | ADP | NA | NA |
| 7 | 3 | 2 | 1 | Queda por ver cómo se consigue articular y qué tal se... | 3 | ver | ver | VERB | NA | VerbForm=Inf |
| 8 | 3 | 2 | 1 | Queda por ver cómo se consigue articular y qué tal se... | 4 | cómo | cómo | ADV | NA | NA |
| 9 | 3 | 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 | 3 | 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 |

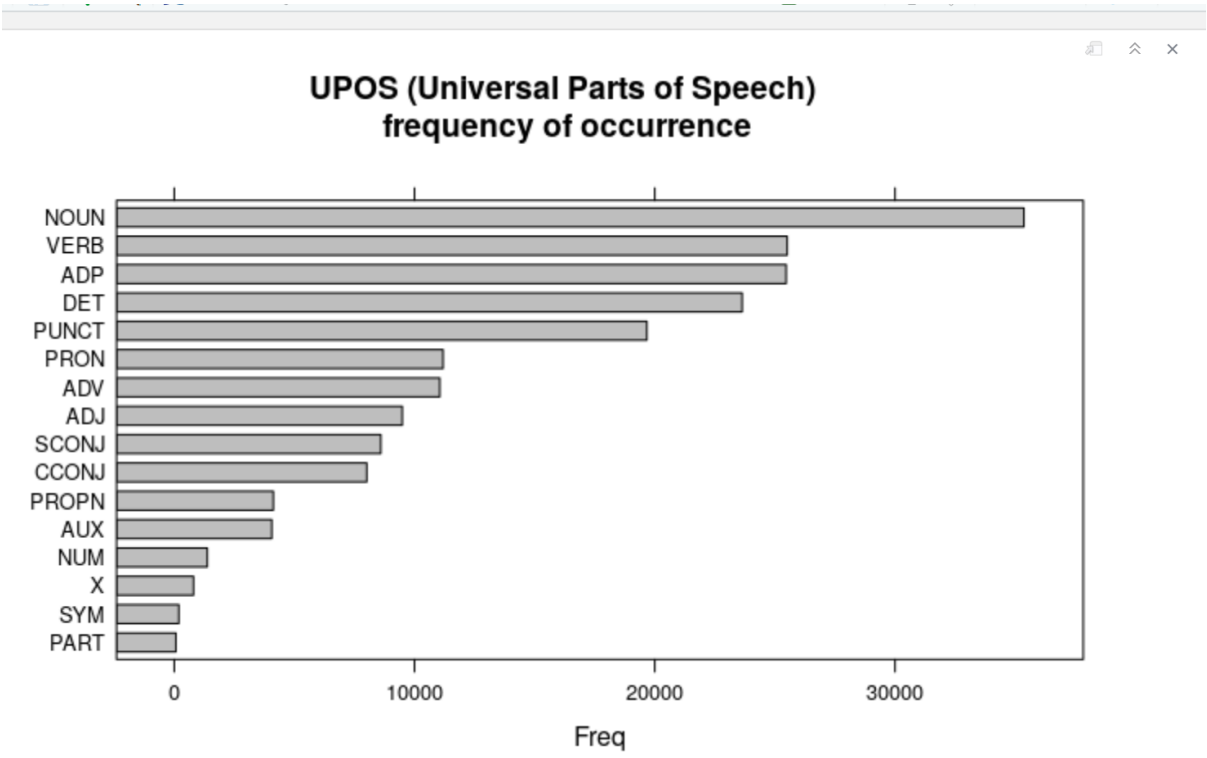
**Figura 14.** *Resultado del análisis del algoritmo utilizado por UDPipe para tokenizar y anotar el texto de los comentarios ciudadanos.*

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.



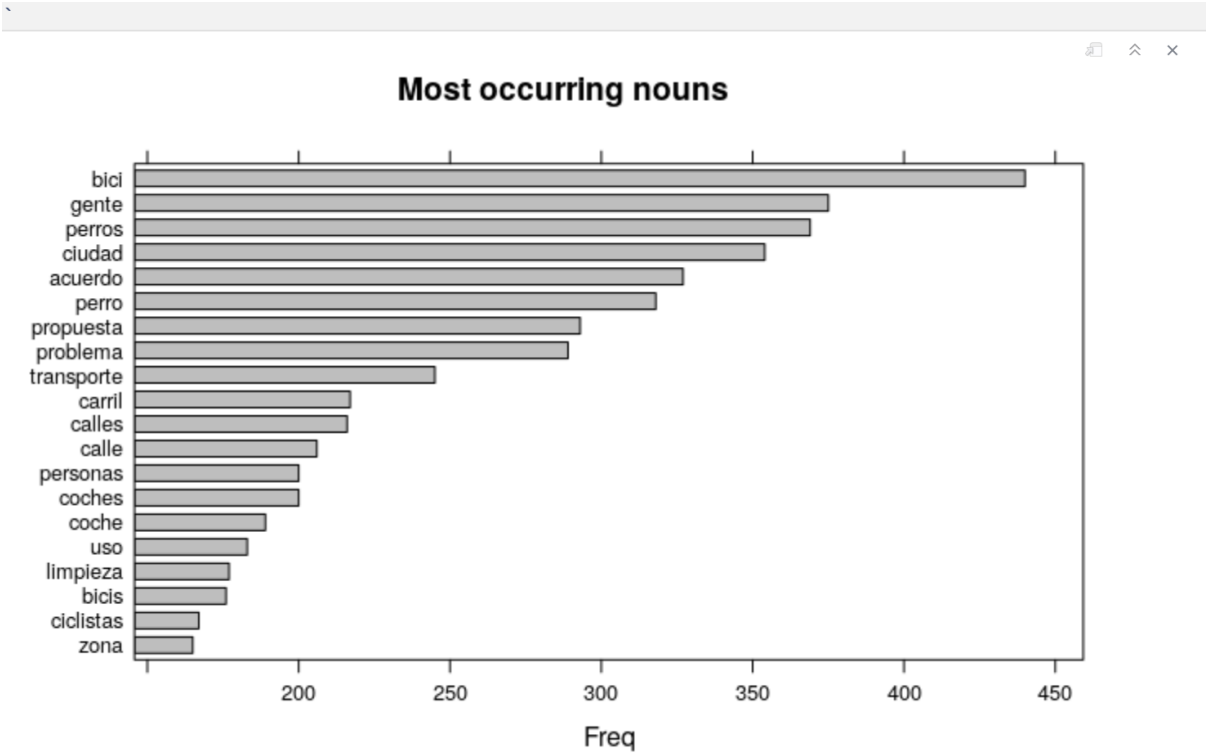



**Figura 15.** *Representación gráfica de algunos de los indicadores producidos por UDPipe. Panel superior: análisis UPOS (Universal Part of Speech) que indica los tipos de palabras más comunes en el conjunto de datos. Panel inferior: nombres más comunes que aparecen en el conjunto de datos.*

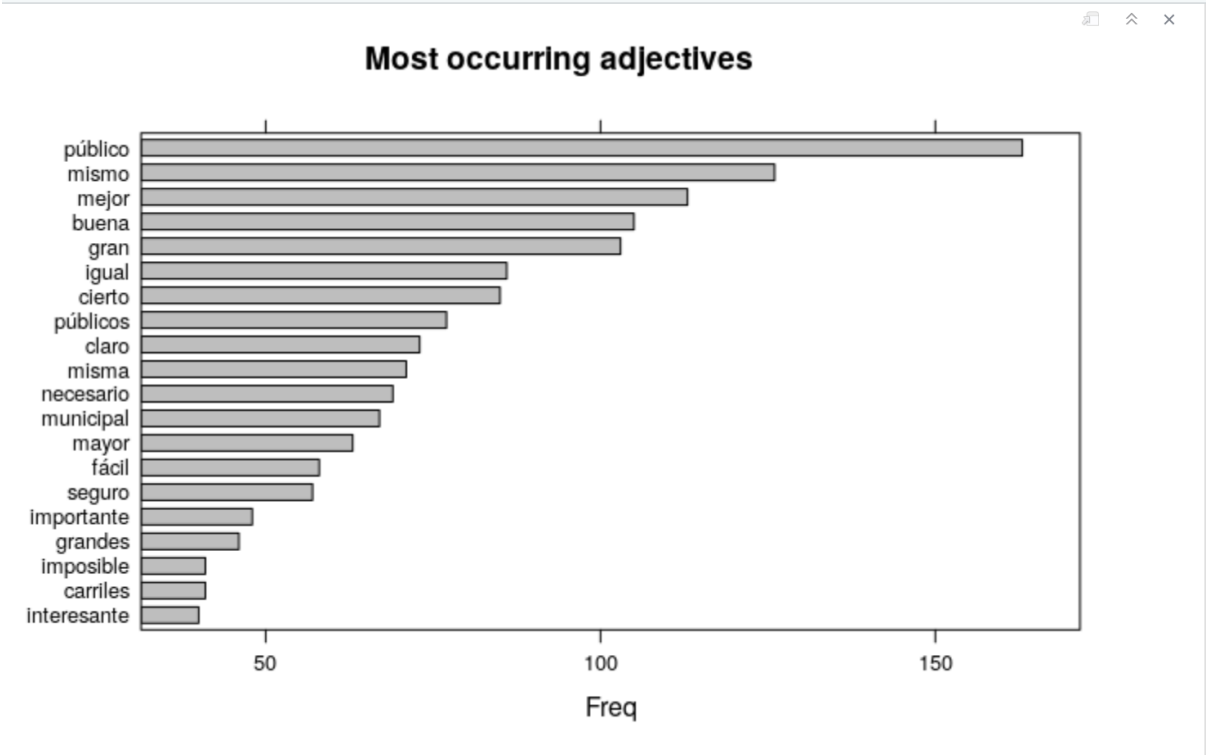


**Figura 16**. *Adjetivos más comunes en el conjunto de datos.*

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")
```

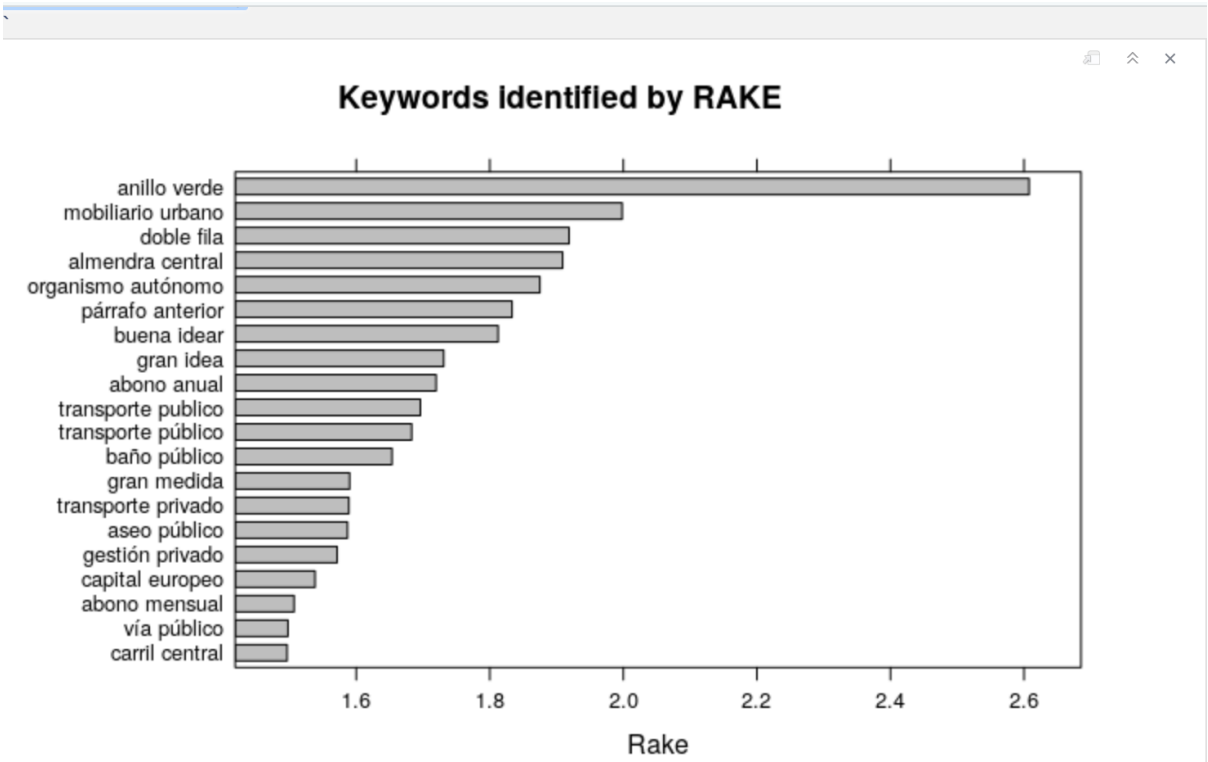



**Figura 17.** *Conjuntos de términos o expresiones clave más comunes en el conjunto de datos de comentarios.*

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.

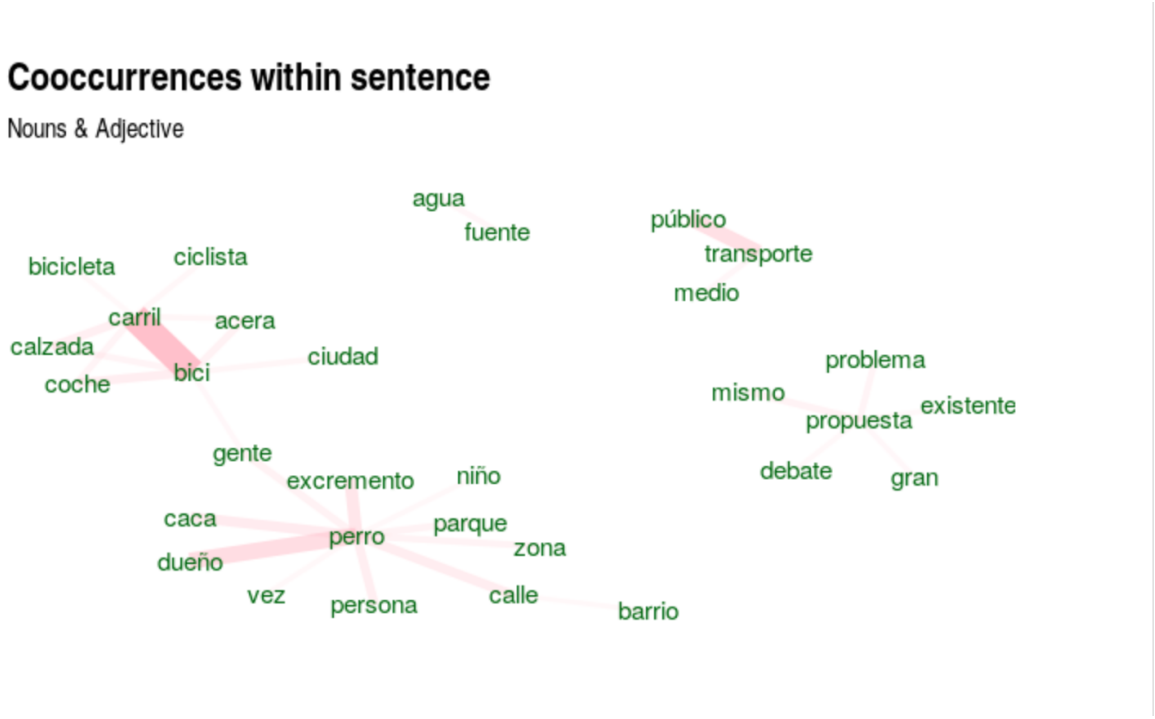



**Figura 18.** *Mapa de conjuntos de términos relacionados clave.*

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
```

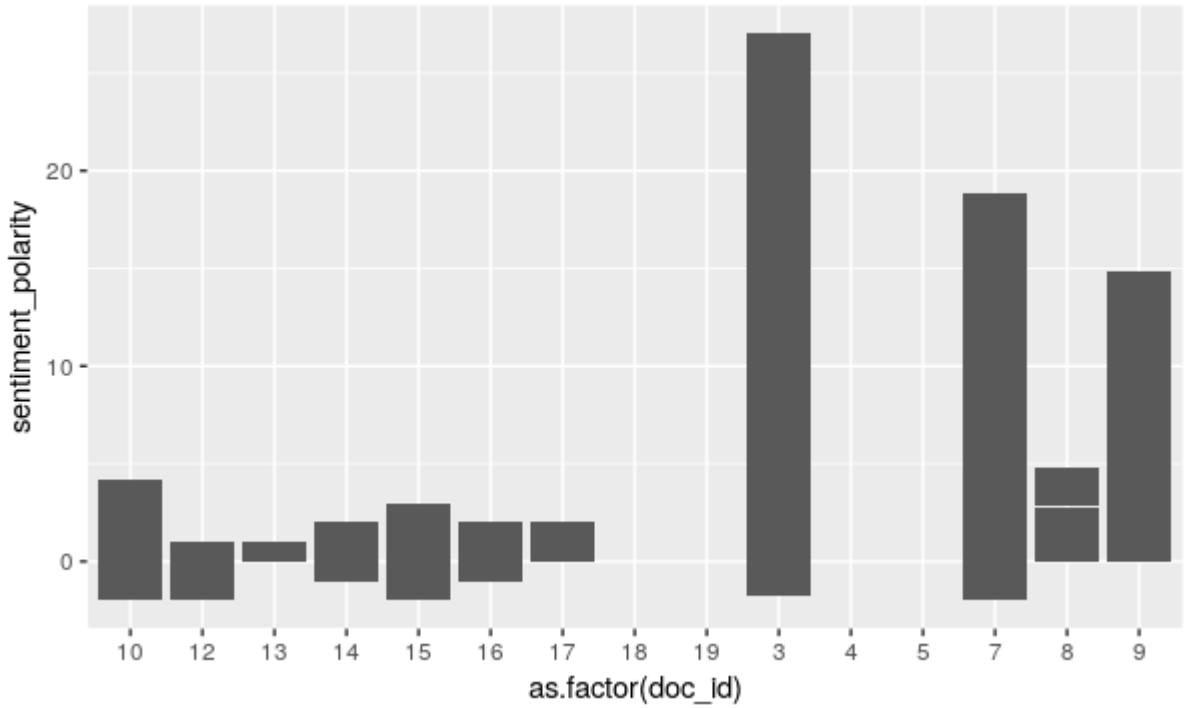



**Figura 20**. *Análisis de sentimiento y polaridad de los debates en base al número de comentarios positivos y negativos.*

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.

# 6. PRÓXIMA PARADA…

Si no has tenido bastante sobre el Procesamiento del Lenguaje Natural. ¡Nos alegramos! Nosotros hemos tenido que parar aquí, pero esperamos que tú no lo hagas. Por eso, a continuación, te dejamos una colección de lecturas muy recomendables para que conviertas en un experto o experta en Procesamiento del Lenguaje Natural.

## 6.1 Colecciones completas sobre NLP

En esta sección podemos encontrar referencias generales sobre NLP que describen de forma introductoria qué es y para que sirve el NLP en nuestros días.

- https://inlpcenter.org/nlp-books-2/
- https://www.sas.com/es_ar/insights/analytics/what-is-natural-language-processing-nlp.html
- https://www.datacentric.es/blog/business/procesamiento-lenguaje-natural-revolucion-futuro/
- https://www.youtube.com/watch?v=5c0qIh54uqE

## 6.2 Word embeddings

En esta sección podemos profundizar en el conocimiento de las técnicas de codificación de texto desde las más sencillas hasta las más complejas.

- https://towardsdatascience.com/introduction-to-word-embeddings-4cf857b12edc
- https://towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526

- https://www.tensorflow.org/tutorials/text/word_embeddings

## 6.3 Herramientas software en R y Python para NLP

Algunas herramientas útiles para programadores que quieran experimentar con Procesamiento del Lenguaje Natural desde la perspectiva de la codificación.

- https://cran.r-project.org/web/views/NaturalLanguageProcessing.html
- https://www.nltk.org/
- https://spacy.io/
- https://medium.com/@srimanikantapalakollu/top-5-natural-language-processing-python-libraries-for-data-scientist-32463d36feae

## 6.4 Aplicaciones prácticas sobre el lenguaje

Algunos casos de uso y aplicaciones prácticas explicadas con detalle.

- https://heartbeat.fritz.ai/a-2019-guide-for-automatic-speech-recognition-f1e1129a141c
- https://blog.insightdatascience.com/how-to-solve-90-of-nlp-problems-a-step-by-step-guide-fda605278e4e

- https://medium.com/@mattkiser/an-introduction-to-natural-language-processing-e0e4d7fa2c1d
- https://www.machinelearningisfun.com/get-the-book/