Analizador de notícias a partir de diferentes fuentes de información

Diego Román Rodríguez

Resumen– En la actualidad, se ha incrementado de manera muy alarmante la difusión de desinformación y en muchos casos, la difusión de noticias tergiversadas de medios de comunicación, que a nuestro parecer, son fiables. El grado de penetración de internet, redes sociales y otro tipo de fuentes digitales de información, han facilitado la creación de un escenario en el cual es muy fácil desinformar, influenciar la opinión pública y afectar la integridad de las personas, condicionados por intereses económicos o políticos. Llegados a este punto, este proyecto nace con el objetivo de mitigar los efectos anteriormente nombrados. Para ello, se propone el uso de modelos de Inteligencia Artificial para realizar un análisis de las noticias, con el objetivo de que el usuario lo pueda usar como una referencia de su información, pudiendo conocer características de las noticias que reflejarán el grado de fiabilidad de las mismas.

Palabras clave– word2vec, similitud vectorial, desinformación, noticias, NLP, POS tagging, NER

Abstract– Nowadays, the dissemination of misinformation has increased very alarmingly and in many cases, the dissemination of distorted news media that in our opinion, are reliable. The degree of Internet penetration, social networks and other types of digital information sources, have allowed the creation of a scenario in which it is very easy to misinform, influence public opinion and affect the integrity of people conditioned by economic or political interests. At this point, this project is born with the aim of mitigating the effects previously mentioned. For this purpose, the use of Artificial Intelligence models is proposed to carry out an analysis of the news with the aim to use it as a reference of user's information, being able to know characteristics of the news that will reflect the reliability degree of them.

Keywords– word2vec, vector similarity, missinformation, news, NLP, POS tagging, NER

1 Introducción

Actualmente, dada la era digital en la que nos encontramos, se ha producido un incremento alarmante de desinformación[1] e incluso de difusión de noticias tergiversadas, las cuales no únicamente representan un efecto producido en las redes sociales, sino que además, no es casual que aparezcan en medios oficiales de comunicación.

El efecto producido por dicha situación, nos adentra en un tipo de problema complejo, el cual se podría abordar de diferentes maneras y en diferentes ámbitos. El ámbito en el cual nos centraremos en este proyecto, es el de las fuentes de información oficiales y en cómo dicha información es mostrada de una determinada manera o perspectiva, que puede inducir al lector a tomar una posición influenciada por la fuente que está consumiendo. Por otro lado, en lo relativo al ámbito, nos centraremos en trabajar noticias de índole político, ya que nuestra sociedad está influenciada y determinada por la relación entre la política y la comunicación [2].

Según un informe de *Digital News Report*[3], los internautas españoles cada vez desconfían más de las noticias en formato digital. Lo que indica que a partir de la tensión social y política vivida durante los últimos años, ha aumentado la desconfianza debido al descrédito institucional de los medios de comunicación. Podemos ver los resultados de la encuesta en la Figura 1.

• E-mail de contacto: diegofernando.roman@e-campus.uab.cat
• Mención realizada: Enginyeria del Software
• Trabajo tutorizado por: Dr. Ramon Baldrich Caselles (Departament de ciències de la computació)
• Curso 2018/19

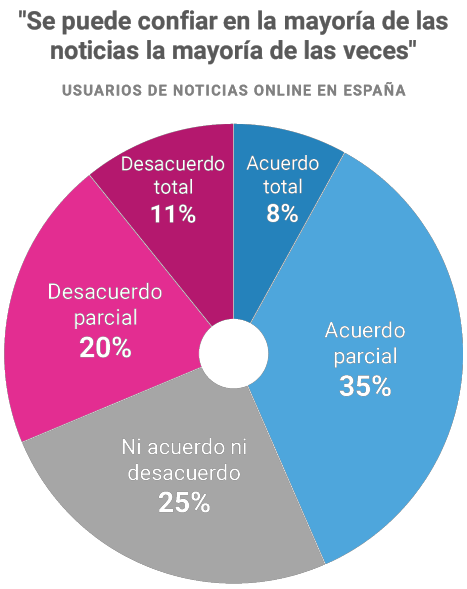

Fig. 1: Confianza de los internautas españoles del 2019

1.1. Motivación

Dado el contexto actual y mi gran interés por la política nacional e internacional, prácticamente cada día consulto fuentes de información. El dato curioso es que, en muchas ocasiones me he visto empujado a pensar de una manera concreta, definida por la subjetividad inherente de las fuentes de información como los diarios, la televisión o la radio. Debido a esto, recientemente empecé a buscar otro tipo de fuentes e incluso a informarme a partir de fuentes de información alternativas para intentar obtener otros puntos de vista, con los cuales poder hacer un contraste de la información. Dicho esto, la dificultad que he podido experimentar ha sido muy alta y nunca acabas de estar convencido de todo lo que lees. A la vez, es muy costoso intentar encontrar fuentes de información que se puedan verificar como objetivas e informativas.

La situación anteriormente expuesta y el estudio realizado por *Digital News Report*, me ha hecho reflexionar y entender que este efecto no es algo específico o aislado, sino que se trata de un efecto general en la sociedad española, y probablemente en la sociedad actual. Dado que es el tipo de trabajo oportuno para investigar y poner en práctica los conocimientos adquiridos en el grado, me he visto casi en la obligación de intentar aportar una solución tecnológica a la problemática existente.

Esta solución pasa por crear un modelo de Inteligencia Artificial[4], el cual facilite unas métricas determinadas para evaluar el contenido de las noticias, su respectiva información y un contraste de la misma noticia obtenida a partir de diferentes fuentes. Así logramos que el usuario, no sólo tenga información del contenido de la noticia, sino que también tendrá información acerca de cómo se le está informando por los medios que consulta. Además, se podrá concluir si la noticia quiere inducir al lector un tipo de postura sobre el objeto de información que podrá inferir a partir de los resultados del análisis.

2 Objetivos

En consecuencia a la motivación descrita en el apartado anterior y a la problemática actual en la era digital, se ha planteado como objetivos principales de este proyecto el desarrollo de un modelo de Machine Learning y una aplicación web que permita al usuario visualizar los resultados del análisis.

2.1. Modelo de Machine Learning

Un modelo de Machine Learning que dadas dos noticias del ámbito político, devuelva los siguientes resultados determinados por la predicción:

- **Diferencia de texto**: Para tener más información acerca de las noticias, es necesario saber en qué se diferencian y qué tipo de protagonismo tiene dicha información. Gracias a ello podremos determinar, si la diferencia entre ambas noticias carece de significado o, por el contrario, tiene relevancia ya que actúa como información adicional para complementar el hilo principal

- **Conjunto de palabras clave de las noticias según el contexto**: Esta funcionalidad permitirá obtener las palabras más importantes de cada una de las noticias y aplicar un contraste de los conjuntos, con el objetivo de determinar las palabras que marcan la diferencia entre dos noticias que, aparentemente, informan del mismo tema.

- **Similitud de las noticias a partir del texto**: Esta funcionalidad nos permitirá obtener la similitud semántica entre dos noticias, las cuales podrían tener un mismo tema general, pero que no informan exactamente de lo mismo. Dicho de otra manera, se quiere mostrar que dos noticias se tratan de una manera diferente según la fuente de información. Es decir, las noticias podrían estar usando el mismo tema como hilo principal para informar, pero enfocándose en otra información relacionada o secundaria. Esta similitud nos permite saber cuán alineadas se encuentran las noticias en cuanto al tema del que están informando.

- **Similitud de las noticias a partir del texto resumido**: Dado que la similitud de las noticias a partir del texto puede tener un contexto distinto, también es necesario identificar si a partir del resumen de las noticias, siguen teniendo el mismo significado. El resumen permite obtener el mensaje más importante de la noticia, sobre el cual se volverá a aplicar la similitud semántica.

2.2. Aplicación web

A partir de dos noticias de diferentes fuentes de información, la aplicación web permitirá al usuario poder interactuar con el algoritmo de Inteligencia Artificial, el cual expuesto a partir de un servicio HTTP, recibirá los enlaces de las noticias elegidos por el usuario y retornará los resultados de forma estructurada, con la cual la aplicación mostrará:

- **Fuente y título de la noticia**: Permite identificar cada una de las noticias para poder compararlas.

- **Contenido de la noticia** con la que el usuario podrá interactuar.
- **Porcentaje de similitud**: Se muestra la similitud de ambas noticias de forma visual, tanto para la noticia entera como para el resumen de la misma.
- **Palabras clave**: Cada noticia tiene un conjunto de palabras clave de su contenido. Serán interactivas y el usuario podrá seleccionarlas para verlas en el texto de la noticia.
- **Resumen**: Permite tener una visión general del contenido de la noticia.
- **Diferencia de texto**: Cada una de las noticias tendrá un texto el cual le diferenciará de la otra noticia. Nos permite poder compararlas a partir de su diferencia.

3 Estado del Arte

En este apartado introduciremos los trabajos ya realizados por otros investigadores, los proyectos que nos servirán como referente en cuanto a solución software y las herramientas con las cuales trabajaremos en este proyecto.

3.1. Trabajos previos

Durante las últimas décadas, muchos investigadores han tenido inquietud para determinar, a partir de análisis algorítmicos, la fidelidad a la realidad de una noticia y el grado de subjetividad que subyace en ella. Debido a la revolución de la Inteligencia Artificial, promovida esencialmente por el aumento de potencia del hardware que se está fabricando, se ha implementado bibliotecas de software que permiten realizar de una manera más cómoda y rápida la construcción de modelos algorítmicos de Machine Learning[5] o Deep Learning[6], con los cuales tratar el texto de las noticias para analizarlo y poder clasificarlo.

Si bien hemos comentado el uso de algoritmos de Inteligencia Artificial, nuestro objetivo es usarlos para el tratamiento de texto, también conocido como NLP (*Natural Language Processing*)[7]. Se trata de un término general utilizado para describir la capacidad de una máquina de procesar texto y entender su significado. Este es un requisito indispensable que deberían tener las bibliotecas que podremos usar para construir el modelo.

Los trabajos realizados hasta la fecha, en su gran mayoría se trata de detectores de *fake news*[8], comparadores de puntuaciones de artículos y de análisis de similitud entre documentos. Los detectores de noticias falsas permiten definir una noticia a partir de características como veracidad o certeza que resulta interesante, pero no se ajusta exactamente al problema que queremos abordar. En cambio, los modelos de análisis de similitud de documentos, usan textos tales como puntuaciones de usuarios a productos o servicios y frases muy cortas, que tiene un grado de complejidad más bajo. También podemos encontrar sistemas recomendadores o sistemas que permiten auto-completar palabras en función del contexto[9].

3.2. Proyectos

Unas de las iniciativas sobre las cuales basaremos una referencia para realizar éste, son los siguientes:

- **Fakebox**: Se trata de un proyecto realizado por una empresa privada, el cual analiza noticias para determinar en qué grado son reales o no. Para ello, se realiza un análisis del título, el contenido y el enlace[10]. Proporciona utilidades tales como: advertencia a los usuarios antes de compartir contenido cuestionable en una plataforma o bien asegurarse de que el contenido del documento es imparcial y su título no es *clickbait* [1]. Podemos encontrar más datos acerca de su implementación y del desarrollo del modelo en el siguiente enlace: fake news detection AI.
- **SenticNet**: Es una iniciativa de investigación concebida en el MIT Laboratory en 2009. Desde entonces, ha promovido el desarrollo y el diseño de sistemas inteligentes sensibles a la emoción en campos como la interacción persona-máquina. El objetivo más importante es conseguir que la información conceptual y afectiva transmitida por el lenguaje natural sea fácilmente más accesible para las máquinas[11].
- **Sentiment Analysis**: Existe una gran variedad de proyectos que permiten realizar un análisis de sentimiento a partir de comentarios en la red, opiniones de los clientes o redes sociales. Este tratamiento computacional se usa de forma más intensiva en áreas de marketing o comunicación con el fin de elaborar estrategias de marketing más elaboradas y efectivas debido al grado de personalización que permite este tipo de modelos. Además, también se suele usar estas herramientas con el fin de conocer la ideología política, la tendencia de voto, etc. [12].

3.3. Herramientas

En este apartado, se hará una breve explicación de las herramientas que usaremos y sus funciones, así como de la aportación al proyecto que se va a desarrollar.

3.3.1. Word2Vec

Se trata de un modelo que permite representar palabras como un vector en un espacio multidimensional de manera que las palabras cercanas o similares tengan puntos del vector también cercanos. Así es posible, no solo capturar la proximidad que hay entre las palabras, sino también la proximidad semántica de ellas. El modelo *Word2Vec*[13] se encuentra disponible en dos formas: *Continuous Bag-of-Words (CBOW)* o el modelo *Skip-Gram*. *Skip-Gram* suele funcionar mejor que *CBOW*[2] sobretodo en conjuntos de datos muy grandes. El modelo usa una gran cantidad de texto para ser entrenado el cual es llamado *corpus*. Dado un *corpus*, el modelo analiza las palabras de cada frase y trata de usar cada palabra para predecir que palabras serán vecinas con la mayor probabilidad debido a la relación que hay entre ellas dentro de un contexto determinado.

[1] Cibercebo - Describe a los contenidos en Internet que apuntan a generar ingresos publicitarios.

[2] Continuous bag of words

- **CBOW**: Dado un conjunto de palabras (*corpus*), el algoritmo analiza las palabras de cada frase e intenta usar cada palabra para predecir qué palabras serán vecinas de ésta.
- **Skip-Gram**: Permite predecir el contexto dada una palabra. Es decir, en este caso, podremos predecir una palabra que precederá la palabra dada, en función de la similitud que tengan.

Este modelo nos permitirá crear un conjunto de *word-embeddings*[14] a partir de las noticias relacionadas con la política. Existen varias bibliotecas de código abierto que tienen una implementación de este modelo en sus dos variantes. Para ello, usaremos la biblioteca **Gensim**[15].

3.3.2. Spacy

Es una librería que permite realizar procesamiento de lenguaje natural (NLP[3]) sobre Python, diseñada específicamente con el objetivo de ser una biblioteca útil para implementar sistemas listos para producción[16]. Debido a que es una librería que permite el uso de *word-embeddings* externos, se adapta perfectamente a nuestros objetivos. Sus funcionalidades más destacadas son:

- **POS (part of speech) Tagging**: Identificación del tipo de palabra en función del contexto.
- **Named Entity Recognition**: Permite identificar las entidades de un texto a partir de un modelo ya entrenado.
- **Sentence Segmentation**: Dado un texto (una noticia), permite su división en diferentes frases.

4 Metodología

La metodología seleccionada para planificar el proyecto será KANBAN. Se ha escogido debido a que permite visualizar las tareas de una manera muy sencilla sobre un flujo de trabajo determinado.

Para obtener el mejor ajuste del proyecto y en función de cada una de las tareas establecidas, se ha fijado un WIP[4] de 2. Se ha determinado así, debido a que las tareas relacionadas con la construcción del modelo de predicción tienen un proceso automático, el cual será realizado por la máquina y en el cual se podrá dedicar el tiempo a otras tareas. De esta manera, se puede aprovechar mejor los tiempos en los cuales no se puede trabajar en una tarea, para dedicarlo a otra.

El flujo de trabajo consta de las siguientes etapas:

- **Por hacer**: Tareas que están a la espera de ser empezadas.
- **En progreso**: Tareas que están siendo trabajadas en la actualidad.
- **Hecho**: En esta etapa se encontrarán las tareas que se ha planificado y que dentro del tiempo y fechas estimadas se finalizaron.

[3]Natural Language Processing
[4]Work In Progress

El proyecto consta de tres módulos los cuales se encuentran planificados sobre el mismo panel:

- Modelo de Inteligencia Artificial.
- Desarrollo de API[5] que expondrá los modelos en un servicio web.
- Desarrollo de aplicación web que consumirá los servicios brindados por la API.

La priorización de las tareas se ha hecho según la importancia que tienen para la realización del proyecto. En este caso, se designa como más importantes las tareas relacionadas con la implementación del modelo de predicción. Dado que es un modelo analítico, se ha de realizar una serie de pruebas para validar el modelo y conseguir ajustarlo en función de los resultados obtenidos y que demos por buenos. Como se ha comentado anteriormente, esta fase es crítica debido a que se trata del núcleo del negocio de la aplicación.

5 Trabajo realizado

En este apartado se explicará cada una de las etapas llevadas a cabo para consolidar la realización del proyecto.

5.1. Arquitectura

En este apartado se realizará una introducción a la arquitectura de la aplicación y su funcionamiento.

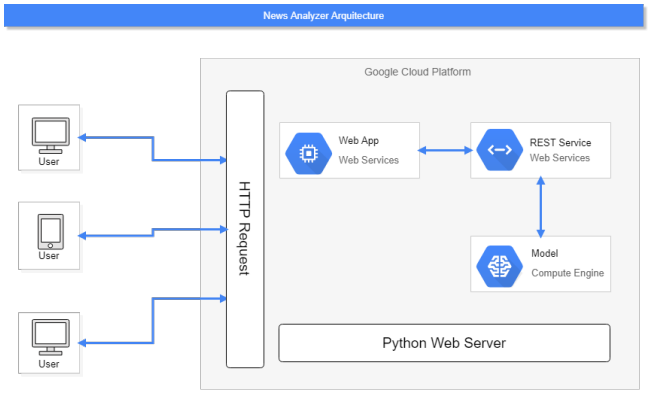

Fig. 2: Arquitectura de la aplicación

Como podemos ver en la Figura 2, la aplicación cuenta con 3 módulos:

- **Web App**: Aplicación web desde la cual el usuario podrá solicitar los análisis de las noticias a partir de enlaces. Además, esta aplicación mostrará al usuario el resultado del análisis una vez se haya procesado.
- **REST[6] Services**: Se trata de un servicio que permite instanciar al modelo de Machine Learning para procesar la solicitud del usuario. También se encargará de obtener el resultado del modelo y devolverlo a la interfaz web en un formato estructurado.
- **Model**: Modelo de Machine Learning que realizará las diferentes operaciones analíticas para obtener los resultados que se mostrarán en la interfaz web.

[5]Application Program Interface
[6]Respresentational State Transfer

5.2. Creación del conjunto de datos

Para dotar de datos de entrenamiento el modelo que queremos crear, es necesaria la creación de un conjunto de datos con el cual poder trabajar. Para ello, se ha procedido a la creación de una aplicación en Python que contiene funcionalidades para la obtención de noticias a partir de sitios web. Está separada en dos módulos:

- **Web Scraper**: Herramientas que nos permiten obtener datos de las páginas web. Debido a que cada sitio web tiene la información estructurada de una manera distinta; se creó un *scraper* para cada uno de ellos. Uno de los inconvenientes que tuvimos que superar fue el de la carga de contenido dinámico de los sitios web, por lo cual, la complejidad de obtención de los datos aumentó. Este código nos permitió obtener todos los enlaces a noticias con un filtro determinado. Los enlaces se guardaron en un archivo para ser posteriormente procesados.

- **Article Scraper**: A partir de los archivos de enlaces obtenidos por los *web scrapers*, se realizó la obtención del texto de las noticias para cada uno de los enlaces. Estas noticias se guardaron automáticamente en un archivo, el cual sería nuestro *corpus* para la creación de los *word-embeddings*.

5.3. Diagrama de casos de uso

Se ha realizado un diagrama de casos de uso donde se puede apreciar las distintas funciones que realizará cada actor.

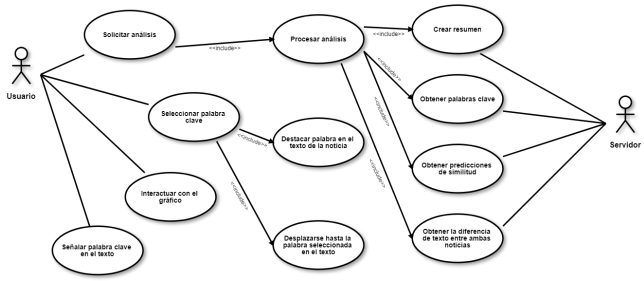

Fig. 3: Diagrama de casos de uso

Como se puede apreciar en la Figura 3, el usuario que interactúe con la aplicación podrá realizar las siguientes operaciones:

- **Solicitar análisis**: Gracias a esta operación, el sistema se encargará de recibir la solicitud que será procesada para mostrar la pantalla de visualización de los resultados. El sistema se encargará de realizar las operaciones pertinentes.

- **Seleccionar palabra clave**: Una vez el usuario seleccione una palabra clave, el sistema se encargará de destacar la palabra seleccionada dentro de la noticia y de deslizar la pantalla para que el usuario pueda verla.

- **Interactuar con el gráfico**: El usuario podrá interactuar con el gráfico que visualiza las palabras clave y el número de ocurrencias dentro de la noticia.

- **Señalar palabra clave en el texto**: Cada vez que el usuario pase el ratón por encima de una de las palabras destacadas, se destacará también el contexto de la misma. De esta manera, también podrá tener información acerca del contexto en el cual se encuentra dicha palabra clave.

5.4. Resumen de las noticias

El resumen de la noticia es imprescindible dado a que no sólo estamos obteniendo la similitud del texto completo de un par de noticias, sino que además, con el resumen tenemos la ventaja de obtener los fragmentos más importantes de cada noticia para realizar también la medición de similitud sobre las mismas, dándole así más sentido al resultado de similitud obtenido.

Con este paso se intenta demostrar que, aunque dos noticias tengan la tendencia de hablar sobre el mismo asunto, y en general el hilo conductor sea el mismo; pueden dar más importancia a fragmentos que no necesariamente son los mismos, con lo cual se está demostrando que el medio o el autor tiene una posición sobre la información y se transmite de una manera más subjetiva. Dada la premisa mencionada anteriormente, es necesario que el resumen del texto sea coherente y tenga sentido. Para realizarlo, se ha creado un algoritmo que se comporta de la siguiente manera tal y como podemos ver en la Figura 4.

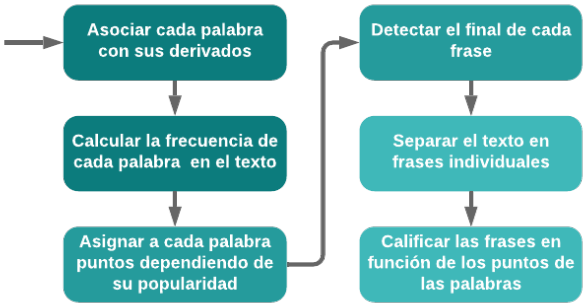

Fig. 4: Proceso de resumen de una noticia

A partir del método de prueba y error, se ha hallado una heurística con la cual determinar el número de frases con las cuales definir el resumen de la noticia. Este número se ha fijado en 5.

Una vez se ha asignado la puntuación a cada frase de la noticia, se retorna las 5 primeras frases con la puntuación más alta, las cuales constituyen el resumen y servirá para posteriormente aplicar el algoritmo de similitud.

5.5. Obtención de palabras clave del texto

Para obtener las palabras clave del texto se usó la biblioteca Spacy, que contiene dos funcionalidades que nos permiten realizar:

1. **POS (part of speech) tagging**: En esta fase se realiza un reconocimiento de cada una de las palabras del texto y se asigna el tipo de palabra que representan en función del contexto[17].

2. **NER (Named Entity Recognition)**: A partir del reconocimiento de las palabras, se procede a realizar la obtención de las entidades dentro del texto, las cuales son un objeto del mundo real que tienen sentido propio aún sin el contexto[18]. En nuestro caso, para cada uno de las noticias se obtiene las entidades con más ocurrencias dentro del texto y se obtiene las 5 primeras. Estas serán las palabras clave de cada noticia. En este proceso, también se fijó el número de palabras clave a partir del método prueba y error con el cual se pudo obtener el número de palabras más adecuado.

Estas entidades tienen un peso fundamental dentro del texto. Para averiguar su importancia obtenemos la frecuencia de aparición de estas entidades en las noticias. Lo hacemos así porque entendemos que el número de ocurrencias de estas entidades dentro del texto, será proporcional al número de frases en las que aparece la entidad en la noticia. Por lo cual, mientras mayor es la frecuencia que tiene la entidad en el texto, más importancia tendrá.

5.6. Diferencias entre las noticias

Para obtener la diferencia de contenido que hay entre las noticias, se ha procedido a usar el conjunto de palabras clave obtenidas de cada uno de los textos que se recibe como entrada tal y como se explica en la fase anterior. A partir de los conjuntos de palabras clave de cada noticia, se procederá a aplicar la siguiente expresión :

$$a = A \setminus B = \{x \in A | x \notin B\}$$

$$b = B \setminus A = \{x \in B | x \notin A\}$$

Donde dadas dos noticias:

A es el conjunto de palabras clave de una noticia y B es el conjunto de la otra. a es el subconjunto de palabras clave conformado por todas las palabras x que pertenecen al conjunto A y que no pertenecen al conjunto B.

Una vez se obtiene ambos subconjuntos, buscamos en cada texto de la noticia las frases en las cuales se encuentra cada palabra de los subconjuntos. El conjunto de frases resultante de cada texto, denotará la diferencia que hay entre las noticias.

5.7. Modelo Word2Vec

Aprovechando la capacidad de cómputo de las máquinas que se fabrican en la actualidad, se ha seleccionado este modelo debido a que permite explicar el sentido y la relación que tienen las palabras en un contexto determinado. Gracias a ello será posible modelar el mundo de las noticias para valorar como son de diferentes y porqué.

Se trata de un modelo de una red neuronal con una única capa oculta. El modelo se entrena con el conjunto de datos de noticias obtenido gracias a las herramientas de *Scraping*[19] implementadas anteriormente. Este modelo no se usará de la manera en cómo se usa los modelos ya entrenados de Machine Learning. En nuestro caso, el objetivo es aprender los pesos de la capa oculta, los cuales son los *word-embeddings*[7] que necesitamos para codificar las palabras de una noticia.

Ahora bien, para entender cómo se ha entrenado este modelo, primero debemos plantear la siguiente cuestión: ¿Dada una palabra, puedo predecir su contexto?

Usamos el modelo *Word2Vec* con su variante *Skip-gram*, que recibe como entrada todo el *corpus* separado en conjuntos de palabras de tamaño determinado, llamado *window*. Este parámetro se suele fijar con la inicialización del entrenamiento del modelo. El modelo capturará información de cada palabra a partir de su contexto y podrá asignar los pesos de la capa oculta. En la capa oculta existe una neurona por cada una de las palabras del vocabulario del *corpus*. Esta variante permite capturar la información tanto semántica como sintáctica, que es necesario para generar unos *word-embeddings* de mayor calidad para cumplir el objetivo que se ha marcado.

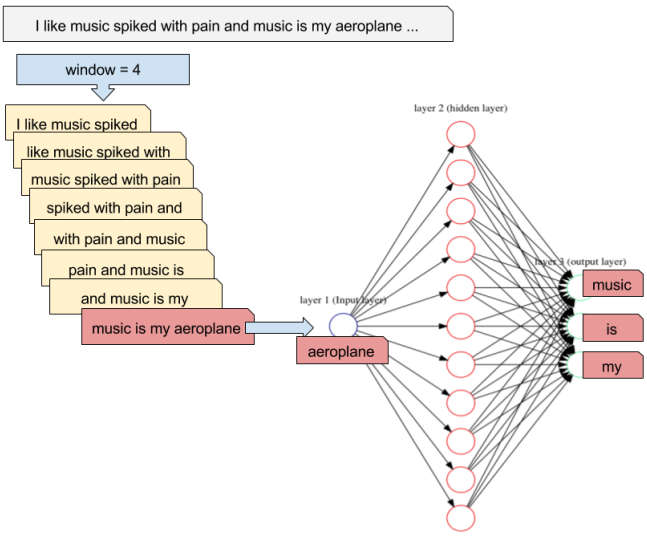

Fig. 5: Arquitectura del modelo Word2Vec [20]

Como podemos ver en la Figura 5, como entrada de la red neuronal se usa un conjunto de palabras, el cual contendrá tantas palabras como se haya fijado en el parámetro *window*. Esta ventana se irá deslizando y se usará una de las palabras de la ventana como objetivo y las otras como contexto. La palabra objetivo se usará como entrada de la red neuronal, y las palabras del contexto como salida.

Este modelo permitirá generar los *word-embeddings* necesarios para la codificación de un documento (noticia) en un vector, que posteriormente usaremos para calcular la similitud de las noticias representadas como un vector multidimensional.

5.8. Cálculo de similitud de las noticias

Una vez se ha entrenado el modelo, explicado en la fase anterior, procederemos a calcular la similitud a partir de dos noticias.

Como podemos ver en la Figura 6, a partir del texto de dos noticias, en este caso *A* y *B*, procedemos a realizar la extracción de *stopwords* y a convertir el texto resultante en un vector de *word-embeddings* donde cada palabra tiene su

[7] Representación de una palabra como un vector numérico de N dimensiones.

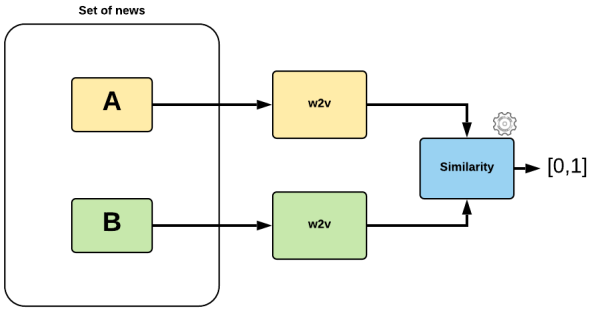

Fig. 6: Proceso de cálculo de la similitud dadas dos noticias

propia representación como un vector de números reales, obtenidos del modelo *Word2Vec* entrenado anteriormente. Una vez tenemos las noticias en su representación vectorial, procedemos a calcular la similitud. El resultado no es más que un valor entre 0 y 1, lo cual nos permite saber cómo de similares son ambas noticias.

Las *stopwords*, también conocidas como ”palabras vacías”son un tipo de palabras que no tienen ningún significado dentro de un texto.

6 Resultados

En este apartado se explicará cómo se ha hecho la validación tanto del modelo *Word2Vec* resultante a partir del entrenamiento con los datos obtenidos, así como los resultados que se ha obtenido tanto del modelo como de la similitud de las noticias. Además, también se mostrará los resultados de la aplicación web que se ha desarrollado para mostrar los datos del análisis de las noticias.

6.1. Validación del modelo Word2Vec

El modelo creado es necesario debido a que los modelos existentes hoy en día son muy generalistas. Han sido entrenados sobre *corpus* como la *Wikipedia*, libros, *Facebook* o noticias, los cuales no son suficientemente adecuados para realizar las predicciones marcadas en este proyecto. Para saber cómo se comportaban estos *word-embeddings* de modelos ya entrenados y cuál era el resultado de las predicciones, se realizó las siguientes pruebas:

Prueba	**Similitud Word-embeddings**	**Similitud (percepción humana)**
Misma noticia	90 %	76 %
Noticias diferentes	90 %	10 %
Noticias relacionadas	88 %	30 %

TABLA 1: Resultados con **word-embeddings** generalistas.

Como podemos apreciar en los resultados de la Tabla: 1, obtenemos unas predicciones de similitud muy elevadas, lo que nos hace intuir que estos resultados son causados debido a la generalización del modelo.

Se ha reentrenado el modelo *Word2Vec* varias veces con diferentes parámetros hasta llegar a obtener uno que haya generado unos *word-embeddings* que expliquen lo más acertadamente la relación que hay entre las palabras con las que fue entrenado. Los parámetros de entrenamiento del modelo definitivo son los siguientes:

Algoritmo	**Épocas**	**Dimensión**	**Ventana**
Skip-gram	120	260	12

TABLA 2: Parámetros de entrenamiento del modelo.

Como podemos observar en la Tabla 2, se realizó varios entrenamientos con las dos arquitecturas pero obtuvimos mejores resultados con *Skip-gram*. El número de épocas escogido viene dado a partir de realizar diferentes pruebas y obtener buenos resultados con este valor. En cambio, con valores más altos, la calidad de los *word-embeddings*, prácticamente no mejoraba. Para la dimensión es recomendado usar un valor entre 100 y 300. El valor usado arrojó mejores resultados, ya que con un valor cercano a 100, la calidad de los *word-embeddings* empeoraba y entendemos que no es suficiente para capturar toda la información de las noticias. Para la ventana, se recomienda valores bajos. En nuestro caso, debido a que las frases de las noticias tienen una longitud bastante mayor, se ha aumentado este valor acorde a ello, para así tener un contexto lo suficientemente grande para capturar toda la información posible de las palabras en un contexto determinado.

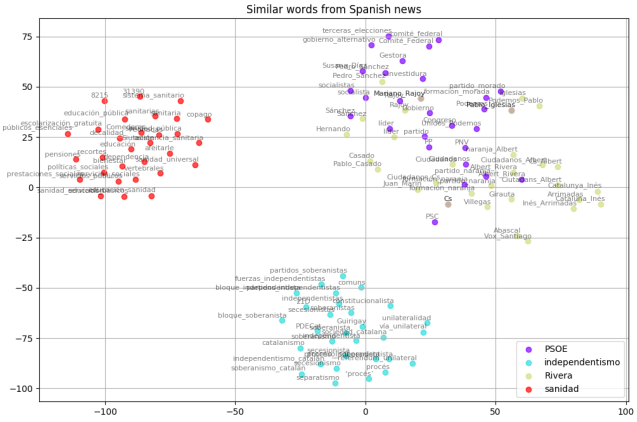

Fig. 7: Clústeres de palabras con sus vecinos más próximos

En la Figura 7, podemos ver una representación gráfica de la distribución de las 20 palabras más cercanas a las vistas en la leyenda. Se puede apreciar que dependiendo de la temática y del contexto del que han sido extraídas las palabras, se forman grupos más o menos independientes entre sí. Para realizar este gráfico se usó t-SNE [8], que permite la visualización de datos con una dimensionalidad muy grande.

6.2. Predicción de similitud

La predicción de la similitud es la parte más importante del proyecto. Una vez el modelo *Word2Vec* ha sido validado, lo usaremos para crear un vector de palabras representadas como vectores y a su vez extraídas del modelo anteriormente entrenado. De esta manera, tendremos una noticia

[8] t-Distributed Stochastic Neighbor Embedding

representada en valores numéricos que permitirá calcular la similitud.

Como pudimos ver en la Figura 6, una vez tenemos ambas noticias representadas como vectores, el paso que queda es realizar el proceso de cálculo de la similitud. Para ello, usamos la distancia coseno entre los dos vectores resultantes:

$$similarity(A, B) = \cos(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Para tener otro punto de vista de cómo interpretar esta similitud y verificar su validez, se ha recreado una representación gráfica en un mapa de calor de una noticia de diferentes fuentes. Para ello, se ha hecho una reducción de las dimensiones del vector usando PCA[9]. Se ha pasado de las 260 dimensiones que representan cada palabra, a tan solo dos.

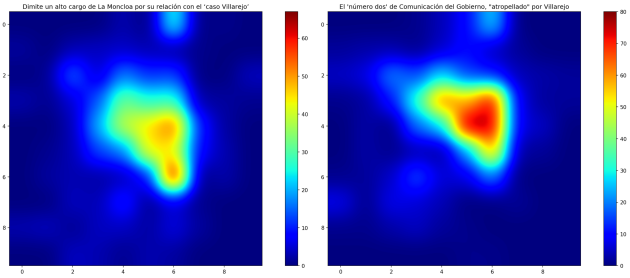

Fig. 8: Misma noticia y diferentes fuentes, representada como un mapa de calor.

Como vemos en la Figura 8, la forma del mapa de calor generado para cada una de ellas es similar. Esto denota que de alguna manera estas noticias sí tienen un tema en común. También se puede apreciar que la gráfica de la derecha, crea un pico con más densidad en el cual se concentra la gran mayoría de palabras. En cambio, la otra concentra estas palabras de una forma más distribuida creando así dos zonas más densas.

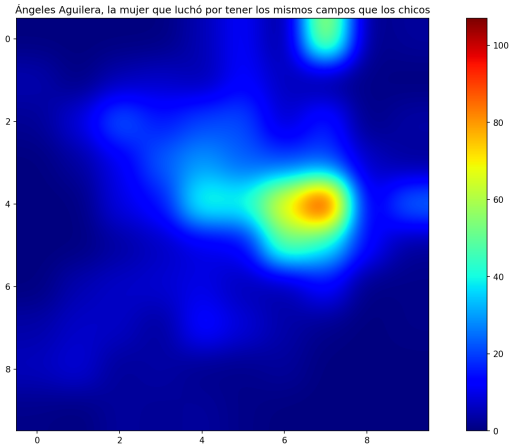

Fig. 9: Noticia de deportes representada como un mapa de calor.

En la Figura 9, se puede ver claramente que la noticia tiene una distribución distinta a partir de su forma y de la densidad.

[9]Principal Component Analysis

A partir de estos mapas de calor de cada una de las noticias podemos hacer una valoración en cuanto a la calidad de los vectores generados. Para acabar, también será necesaria la valoración cuantitativa, que es la que el usuario podrá visualizar en la aplicación. Se ha elaborado la siguiente tabla con los resultados de la similitud obtenida para cada uno de los casos:

Temática	Noticias	General	Resumen
Política	Iguales	93 %	83 %
Política	Relacionadas	91 %	75 %
Política Deportes	Diferentes	67 %	61 %

TABLA 3: RESULTADOS DE SIMILITUD DE DIFERENTES FUENTES POR TEMÁTICA.

Como podemos ver en la Tabla 3, los resultados de similitud de las noticias es elevado, aunque se puede notar una gran diferencia cuando se aplica la similitud a su resumen. En el caso de las noticias que son iguales, la similitud es acorde a la que daría un humano si las llegase a puntuar. En cambio, el resumen sí marca una diferencia más grande. Esto viene dado a que el resumen solo recoge las partes más importantes de la noticia, con lo cual podemos deducir que una de las noticias se centró más a informar sobre un tema, mientras otra profundizó más en otro.

En el caso de las noticias relacionadas, noticias que tiene un tema en común, pero cada una de ellas se centra en informar en un sentido diferente; la similitud es bastante más alta de lo que nos podríamos esperar tanto en el resumen como en la general.

Por último, las similitud de las noticias con temática diferente cae bastante, pero no suficiente como nos esperaríamos. Volviendo a hacer una vista más general sobre los resultados, podemos identificar que aunque los resultados no sean de una similitud parecida a la valoración por inferencia humana, la predicción del modelo sí marca que existe una diferencia entre las noticias que ha recibido como entrada, lo cual significa que el resultado que arroja el modelo no sólo viene dado a partir del contenido de cada una de las noticias y de la temática de ellas, sino que también hay influencia tanto de sintaxis, semántica y del estilo de escritura de las noticias.

6.3. Aplicación web

A partir de los resultados obtenidos del modelo, la aplicación transformará estos datos en dos gráficas tal y como se muestra en las siguientes figuras:

En la Figura 10, podemos ver los porcentajes de similitud referentes a la noticia entera y al resumen de las noticias de las cuales se ha realizado el análisis.

En la Figura 11, podemos ver una gráfica que representa las palabras clave (entidades) más importantes de ambas noticias. Es decir, se muestra las entidades con más importancia que hay en común en las noticias y la respectiva frecuencia con que aparecen. De esta manera, podemos comparar el número de veces que se citan en cada noticia y así poder valorar la importancia que tienen y deducir porqué en cada fuente se informa de una manera diferente.

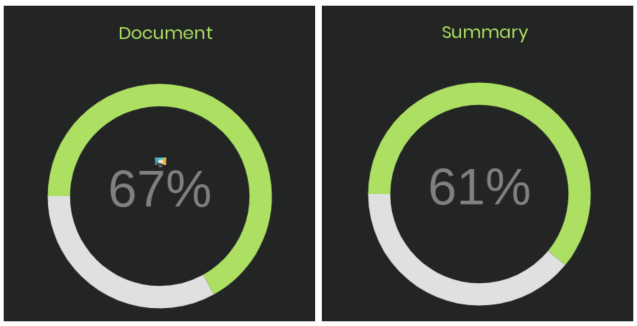

Fig. 10: Similitud de la noticia y del resumen

Fig. 11: Palabras clave más importantes de ambas noticias

7 Conclusiones

Este trabajo ha sido realizado siguiendo la planificación marcada inicialmente. La aplicación funciona correctamente y el usuario final podrá hacer uso de ella sin ningún problema. El uso de las tecnologías ha sido acertado y ha permitido realizar cada una de las tareas, aunque los resultados de la predicción sean sesgados y no permitan obtener con claridad la similitud de las noticias. El conocimiento adquirido durante el desarrollo de este proyecto ha sido notorio y me ha permitido consolidar los conocimientos en lo relacionado a la Inteligencia Artificial y el procesamiento de lenguaje natural. Creo que ha sido un punto clave el hecho de poder combinar un proyecto de Machine Learning con procesos de Ingeniería de software para crear una solución que puede estar lista para desplegar.

Para mejorar la predicción de similitud, seguramente sea necesario realizar un enfoque diferente, usando otros modelos que se puedan ajustar mejor al resultado deseado, explorando otras métricas de cálculo de la similitud y usando más datos de noticias u obteniendo otros de mayor calidad.

Por otro lado, la aplicación web planteada al principio de los objetivos, es usable y se puede interpretar de manera clara los datos que muestra, aunque es muy estática. Posiblemente sea necesario ajustarla para ser más dinámica y que el usuario pueda interactuar con ella de una manera más personalizable.

8 Futuro del proyecto

En cuanto al proyecto en general, creo que ha sido una gran oportunidad el poder realizar la investigación y el aprendizaje en el mundo de Machine Learning y NLP[10]. Por ello, veo una oportunidad el poder seguir trabajando con él para conseguir mejoras :

- Obtener un modelo definitivo con el cual hacer predicciones más acertadas sobre la similitud de las noticias que se recibe como entrada.
- Poder realizar una comparación entre 3 o más noticias.
- Añadir más funcionalidad a la aplicación web para que el usuario pueda interactuar con los datos mostrados y así poder visualizar de manera más sencilla aquello que más le interesa en cada momento.
- Exportación de los resultados a redes sociales.
- Generar una noticia con las noticias recibidas como input entrenando un nuevo modelo con una red neuronal LSTM[11][21].

Agradecimientos

Por último, me gustaría agradecer a todas las personas implicadas en mi trayecto universitario, a las personas que me apoyaron en los momentos difíciles de la etapa y a todos aquellos que me motivaron de una manera u otra para no abandonar mis objetivos. En especial, agradecimientos a mi tutor, por la transferencia de conocimiento durante la carrera y a lo largo de este trabajo; por toda su implicación y su manera de reconducirme para alcanzar con éxito la realización de este proyecto.

Referencias

[1] L. M. R. Rodríguez, *Pragmática de la desinformación : estratagemas e incidencia de la calidad informativa de los medios*. PhD thesis, 2014.

[2] M. Castells, *Comunicación y Poder*. Alianza Editorial, 2009.

[3] A. V. Miguel, "Confianza y desinformación." `http://www.digitalnewsreport.es/2019/report`, 2019. Accessed: 2019-06-12.

[4] V. D. M, "Introduccion a La Inteligencia Artificial," *Instituto Tecnológico de Nuevo Laredo*, 2001.

[5] M. Kubat, *An Introduction to Machine Learning*. 2017.

[6] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[7] T. Beysolow II, *Applied Natural Language Processing with Python*. 2018.

[10] Natural Language Processing
[11] Long-Short Term Memory

[8] N. J. Conroy, V. L. Rubin, and Y. Chen, "Automatic deception detection: Methods for finding fake news," *Proceedings of the Association for Information Science and Technology*, 2015.

[9] G. Semeraro, P. Basile, M. de Gemmis, and P. Lops, "Content-Based Recommendation Services for Personalized Digital Libraries," 2007.

[10] Aaron Edell, "I trained fake news detection AI with >95 % accuracy, and almost went crazy." `https://towardsdatascience.com/i-trained-fake-news-detection-ai`, 2017. Accedido: 2019-03-30.

[11] I. Chaturvedi, E. Cambria, R. E. Welsch, and F. Herrera, "Distinguishing between facts and opinions for sentiment analysis: Survey and challenges," *Information Fusion*, 2018.

[12] D. Mayank, K. Padmanabhan, and K. Pal, "Multi-sentiment Modeling with Scalable Systematic Labeled Data Generation via Word2Vec Clustering," in *IEEE International Conference on Data Mining Workshops, ICDMW*, 2017.

[13] J. Lilleberg, Y. Zhu, and Y. Zhang, "Support vector machines and Word2vec for text classification with semantic features," in *Proceedings of 2015 IEEE 14th International Conference on Cognitive Informatics and Cognitive Computing, ICCI*CC 2015*, 2015.

[14] O. Levy and Y. Goldberg, "Dependency-Based Word Embeddings," 2015.

[15] R. Řehřek and P. Sojka, "Gensim—statistical semantics in python," *statistical semantics; gensim; Python; LDA; SVD*, 2011.

[16] F. N. A. Al Omran and C. Treude, "Choosing an NLP Library for Analyzing Software Documentation: A Systematic Literature Review and a Series of Experiments," in *IEEE International Working Conference on Mining Software Repositories*, 2017.

[17] B. Snyder, T. Naseem, J. Eisenstein, and R. Barzilay, "Unsupervised multilingual learning for POS tagging," 2010.

[18] M. Tkachenko and A. Simanovsky, "Named entity recognition: Exploring features," *Proceedings of KONVENS*, 2012.

[19] E. L. Nylen and P. Wallisch, "Web Scraping," in *Neural Data Science*, 2017.

[20] T. Teofili, "Deep learning for search: Using word2vec." `https://jaxenter.com/deep-learning-search-word2vec-147782.html`, 2016. Accedido: 2019-05-30.

[21] A. Y. Shedko, "Semantic-map-based assistant for creative text generation," in *Procedia Computer Science*, 2018.

ANEXO

A.1. Imágenes de la aplicación Web

En las Figura 12, 13, 14 podemos ver las distintas pantallas de la aplicación web.

A.2. Agrupación de palabras

En la Figura 15, podemos ver las palabras más próximas dadas las palabras situadas en la leyenda. Este gráfico se realizó usando t-SNE [12].

[12] t-Distributed Stochastic Neighbor Embedding

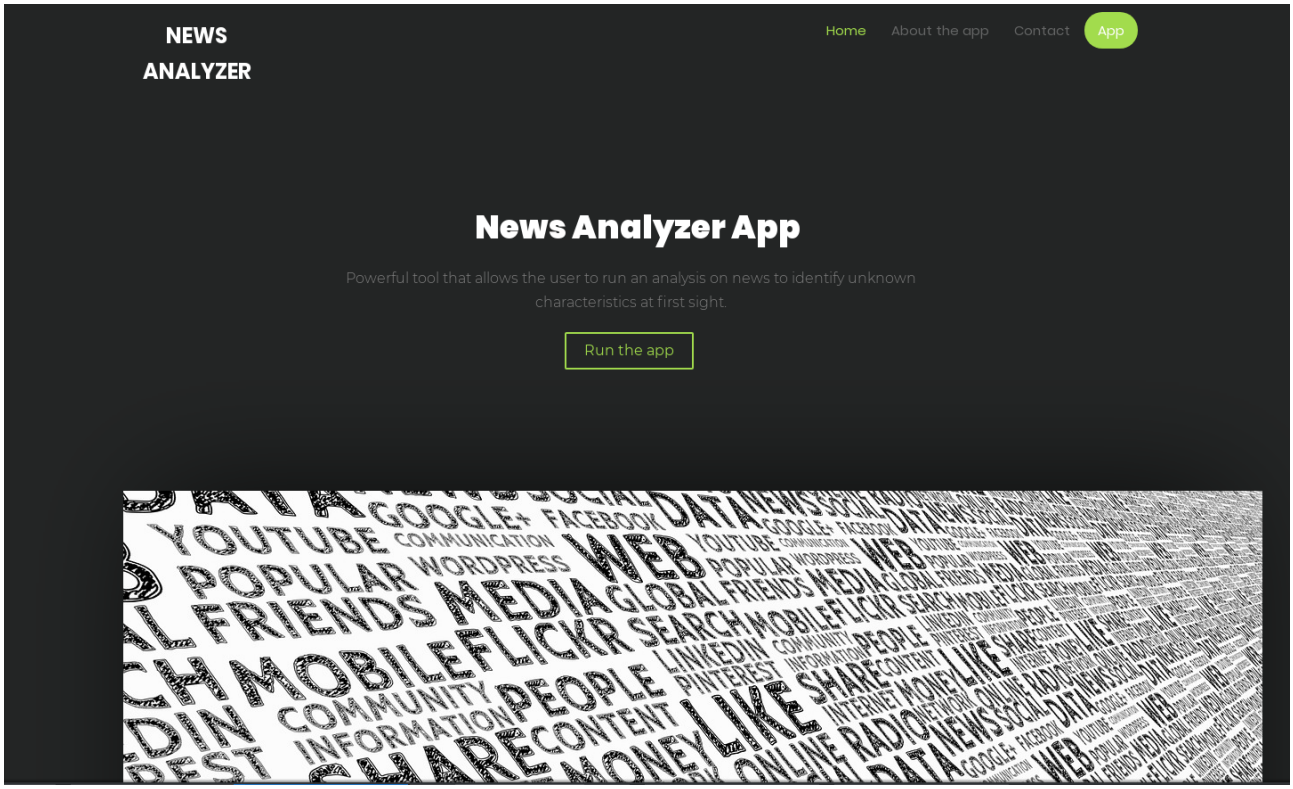

Fig. 12: Página inicial de la aplicación.

Fig. 13: Pantalla de introducción de los enlaces de las noticias que se quiere analizar.

Fig. 14: Resultados del análisis

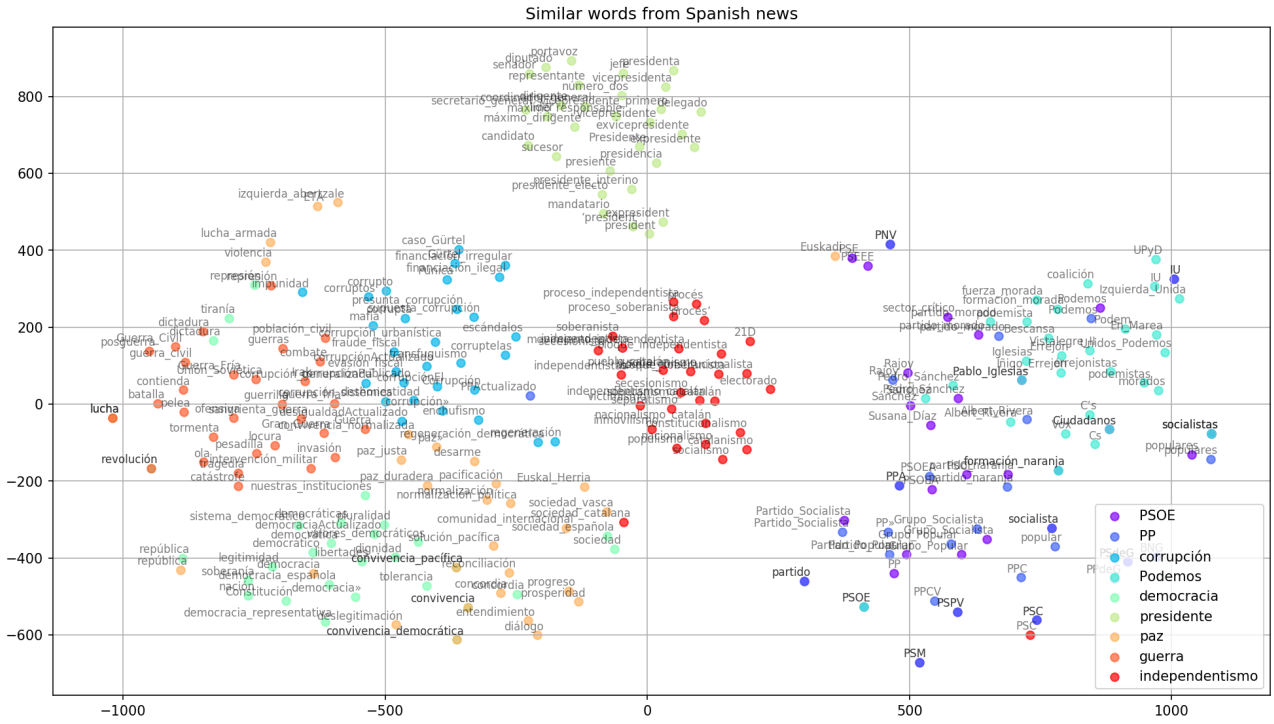

Fig. 15: Clústeres de palabras más próximas