UNIVERSIDAD AUTÓNOMA DE MADRID

Escuela Politécnica Superior

Doble Grado en Ingeniería Informática y Matemáticas

TRABAJO FIN DE GRADO

HERRAMIENTA DE ANÁLISIS Y VISUALIZACIÓN DE NOTICIAS BASADA EN MACHINE LEARNING APLICADO A REDES SOCIALES

Javier del Valle Contreras

Tutor: Simone Santini

Mayo 2018

Herramienta de análisis y visualización de noticias basada en machine learning aplicado a redes sociales

Autor: Javier del Valle Contreras
Tutor: Simone Santini

Escuela Politécnica Superior
Universidad Autónoma de Madrid

Mayo 2018

Resumen

Resumen La cantidad de información que generan las sociedades modernas está en continuo crecimiento. Cuando ocurre un suceso relevante se produce una saturación de noticias de prensa. Esto, sumado a la tendencia de las redes sociales y los buscadores a producir burbujas informativas, puede provocar desinformación. Hace años que la prensa escrita tradicional ha dado paso a un conglomerado de periódicos online, blogs, redes sociales y agregadores de noticias. Actualmente un número creciente de personas accede a las noticias a través de redes sociales como Twitter, Facebook, Whatsapp; o de buscadores como Google y Bing. Estas herramientas ayudan a bucear en el mar de información para encontrar noticias relevantes, pero también pueden introducir sesgos que limitan la exposición a noticias de posiciones contrarias.

En este trabajo se han analizado tres casos de noticias de gran impacto, que han provocado un elevado número de publicaciones en redes sociales. Para realizar dicho análisis, se ha desarrollado una herramienta que permite visualizar el impacto de las noticias de prensa en las redes sociales, e identificar grupos de noticias de ideología similar según la percepción de los usuarios de Twitter.

La herramienta construida permite recolectar un elevado número de tweets relacionados con un suceso de interés, para formar a partir de ellos un grafo de noticias. Posteriormente la herramienta analiza el grafo con diversos algoritmos de detección de comunidades, y ofrece al usuario una visualización interactiva.

Una vez construida la herramienta, se realizaron tres pruebas con diferentes sucesos informativos y usuarios, para evaluar la calidad de los resultados.

El resultado final del trabajo es una herramienta que permite analizar un elevado número de tweets de forma automatizada y proporciona una visualización en forma de grafo que ayuda a identificar las noticias más importantes y los principales grupos de opinión.

Palabras clave Redes sociales, noticias, análisis de grafos

Abstract

Abstract The amount of information that is generated by modern societies is continually growing. When an important event happens, several news is accumulated in the media. Additionally, there is a higher tendency that social media and search engines will create filtered bubbles and echo chambers that cause misleading information. The traditional press has taken a step towards the online press, blogs, social media, and news aggregators. Nowadays, a higher growing number of people are having access to the news through social networks such as Twitter, Facebook, Whatsapp or search engines like Google and Bing. These tools enable individuals to distinguish information that is relevant from other unimportant content. However, these tools can be an open door to bias and will eventually limit the discovery of ideologically diverse opinions.

Through this work, three different news of huge impact has been analyzed. This news has provoked a high number of posts in social media. In order to perform this analysis, a tool has been developed to visualize the impact that that news had on social media. Moreover, this tools can identify groups ideologically similar based on the perspective of Twitter users.

The tool has been built to collect a high number of tweets related to an interesting event and consequently create with them a news graph. Furthermore, it allows analyzing the graph with different community detection algorithms. As a result, the tool offers the user an interactive visualization.

Once the tool was built, three different tests were done with diverse events and users to evaluate the quality of the results.

The final result is a tool that allows the analysis of a large number of tweets in an automatic way. Thus, it provides a visualization of the graph that helps the user to identify the most important news and the main opinion clusters.

Keywords Social media, news, graph analysis

Agradecimientos

Este trabajo no hubiera sido posible sin todas las personas que me han acompañado y guiado a lo largo de estos meses y también durante toda la carrera.

Debo dar las gracias a todos mis compañeros de carrera, que han estado conmigo durante estos años y han hecho mucho más fácil esta etapa de estudios. Quiero agradecer en especial a Cris, por sus buenos consejos y por estar ahí en todo momento.

También quiero agradecer a mi tutor, Simone Santini, por haber aceptado ser mi tutor en este TFG, y por los consejos y ayuda que me ha ofrecido a lo largo de este año.

Quiero agradecer a mi familia, que se sienten orgullosos con todo lo que hago, que me apoyan y me motivan a perseguir lo que me propongo.

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

1 Introducción

1.1 Motivación del proyecto

Desde la aparición de internet el número de periódicos online y blogs se ha multiplicado. Ésto, junto con la irrupción de las redes sociales, ha multiplicado la cantidad de información disponible y ha cambiado la forma en que accedemos a ella. Debido al **exceso de noticias** que se publican cada día, resulta imposible leerlas todas y se debe realizar una selección. Las redes sociales y los buscadores son herramientas muy útiles que permiten bucear en el mar de información para seleccionar los contenidos más relevantes para el usuario. Sin embargo, pueden introducir **sesgos** que invisibilizan opiniones contrarias a la del usuario.

Además, el exceso de medios y la competencia provoca una carrera por la exclusiva, la inmediatez y la cantidad de visitas. Titulares clickbait, noticias no contrastadas, falsas (*fake news*) o información de baja calidad periodística, hacen que cada vez sea más importante informarse desde **varios puntos de vista**.

1.2 Objetivos

Este proyecto busca explorar las posibilidades que ofrece el **análisis de redes sociales** para mejorar la comprensión de las noticias, y enfrentar los crecientes problemas provocados por la sobreinformación y los sesgos en los algoritmos de recomendación de contenidos.

El problema de identificar noticias falsas es también de gran importancia, y está bastante relacionado con los anteriores. Sin embargo, **queda fuera de los objetivos** de este proyecto. Se proporciona bibliografía sobre algunos trabajos que tratan este tema en profundidad [6][7].

El objetivo de este proyecto es desarrollar una **herramienta** que permita analizar y visualizar un conjunto grande de noticias, **agrupándolas** por ideología o afinidad. De esta manera se espera poder facilitar la comprensión del suceso informativo, gracias a poder identificar noticias que traten el tema desde distintos puntos de vista.

La herramienta deberá poder obtener, guardar y procesar un elevado número de tweets, crear un grafo a partir de ellos, y posteriormente analizar el grafo con distintos

algoritmos. El resultado final se ofrecerá al usuario a través de una interfaz gráfica desarrollada con tecnologías web.

Adicionalmente, otro objetivo es el **aprendizaje de conocimientos** relacionados con el análisis de redes sociales, y el manejo de herramientas punteras para ello.

1.3 Desarrollo y plan de trabajo

El desarrollo de este proyecto se ha dividido en tres fases:

- **Investigación y aprendizaje:** Durante esta fase me documenté acerca del impacto de las redes sociales en el periodismo, el funcionamiento de los sistemas de recomendación, y estudios previos de análisis de redes sociales. También me documenté acerca de las herramientas existentes para el análisis de redes y el funcionamiento las APIs que proveen las redes sociales Twitter y Facebook
- **Diseño y desarrollo:** El desarrollo de la herramienta se hizo de manera iterativa. Gracias a las pruebas se pudieron identificar problemas y se fueron resolviendo.
- **Pruebas:** Se realizaron pruebas para evaluar el rendimiento de cada algoritmo, e identificar los más apropiados. Las pruebas se dividen en dos partes: un análisis automatizado utilizando una métrica, y un análisis manual por parte de los usuarios que usaron la interfaz gráfica y que valoraron el proyecto.

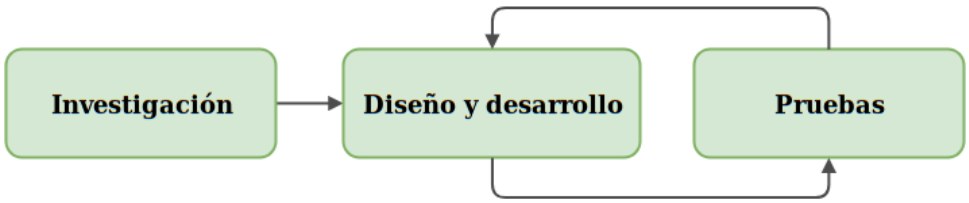

Figura 1.1: Plan de trabajo.

2 Estado del arte

En esta sección se analiza la **situación actual** de los periódicos online, la difusión de noticias y la influencia de las redes sociales en el periodismo.

También se hablará de los principales problemas que introducen las nuevas tecnologías en el acceso a noticias, como son la **sobreinformación** provocada por el gran número de medios online, las **cámaras de eco** en las redes sociales, y los **filtros burbuja** de los buscadores y sistemas de recomendación.

A continuación, se dará una visión general del estado del análisis de redes sociales. Se detallan los **principales algoritmos** de detección de comunidades y cómo han ido evolucionando a lo largo del tiempo.

2.1 Periódicos online frente a medios tradicionales

La irrupción de internet ha cambiado la forma en la que accedemos a la información. En los últimos años la tendencia muestra una caída de los periódicos impresos y una moderada caída de la televisión, a la vez que aumenta el papel de las **redes sociales**.

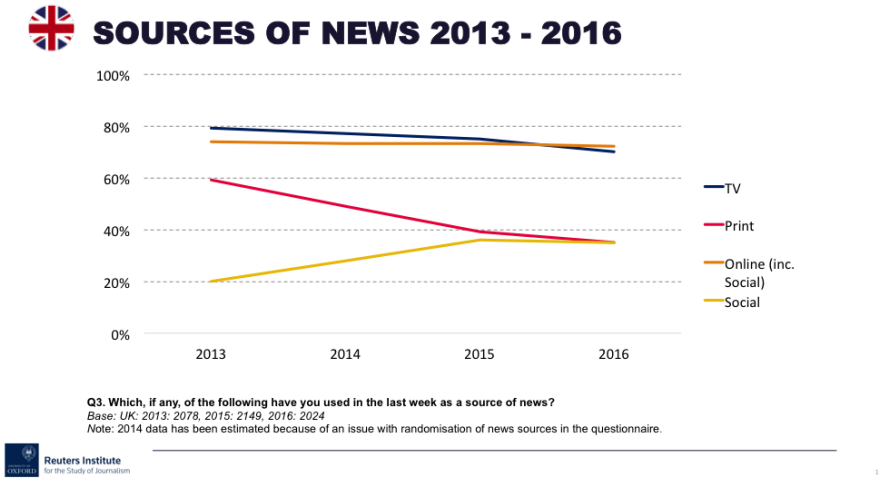

Figura 2.1: Evolución del consumo de noticias a través de los diferentes medios. Imagen obtenida de *Where do people get their news* [1]

Esta tendencia se observa mucho más marcada en usuarios jóvenes entre 18 y 24 años, lo que da una idea del futuro de la difusión de noticias.

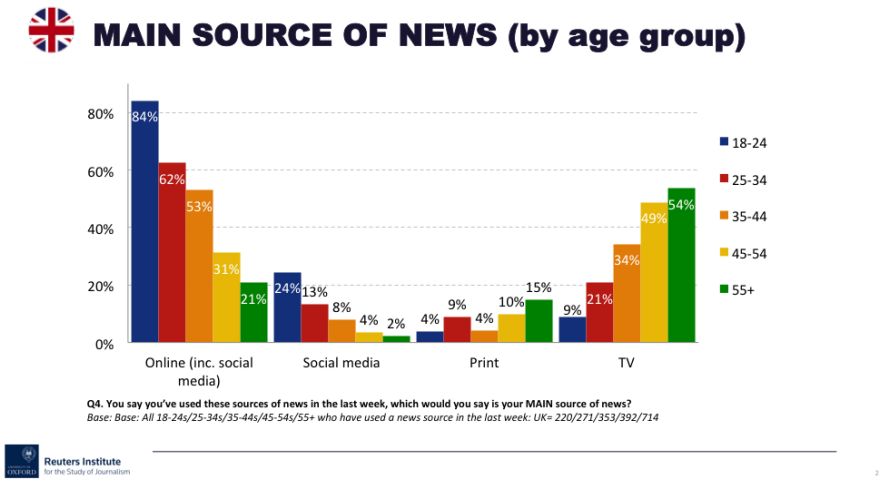

Figura 2.2: Consumo de noticias por grupos de edad. Imagen obtenida de *Where do people get their news* [1]

Los periódicos online tienen ciertas ventajas frente a los tradicionales, entre las que podemos destacar el menor coste de mantenimiento, la eliminación de las restricciones de espacio de las noticias, y la posibilidad de llegar a un número mayor de personas.

En este panorama con un gran número de medios se plantean nuevos problemas para la financiación, y la mayor competencia puede afectar a la calidad de las noticias. Por ejemplo, la disminución de reporteros en el terreno por parte de los periódicos, deriva hacia un modelo de **agencias de noticias** que limita la multiplicidad de versiones. Además, la búsqueda de la inmediatez y la competencia de los distintos medios por ofrecer noticias exclusivas y en tiempo real, reduce el tiempo disponible para contrastar la información y ha llevado a algunos casos muy sonados de *fake news* [6].

2.2 Influencia de las redes sociales en el periodismo

Uno de los cambios más importantes de los nuevos tiempos es el modo en que los lectores **acceden** a las noticias. Si en la era de los medios impresos era muy importante la portada principal, en el mundo actual los lectores llegan a las noticias a través de otras plataformas, principalmente las **redes sociales** [2].

Los medios se han adaptado a este cambio de paradigma y se esfuerzan por posicionar sus perfiles en las redes sociales, y tratar de llegar al máximo número de lectores. Sin embargo, cada vez cobra más importancia la difusión de noticias por parte de usuarios particulares, que comparten opiniones y recomiendan contenidos. Es decir, la comunidad de usuarios ha pasado a tener un **papel activo** en la difusión de noticias [8].

Figura 2.3: Recomendaciones del New York Times para adaptar el periodismo. Las redes sociales toman un papel fundamental. Imagen obtenida de *Diarios impresos vs diarios digitales* [2]

A la hora de juzgar los hechos relatados en una noticia y formar una opinión propia, cobra una gran importancia la opinión de la comunidad sobre ésta. La comunidad añade un contexto a la información relatada por la noticia. Cuando una persona conocida recomienda una noticia, se le tiende a dar más veracidad que a una noticia recomendada por un desconocido [9].

2.2.1 Grupos de opinión en redes sociales. Cámaras de eco

Existen numerosos estudios acerca de las dinámicas sociales que llevan a la creación de **grupos cohesionados**, y cómo éstos afectan a la difusión de ideas e información.

En un artículo publicado en la revista Science [10], Bakshy y Messing analizan el grado de exposición de los usuarios en Facebook a noticias y opiniones alejadas de las propias, concluyendo que el modo en que la información fluye está dominada por la **estructura de conexiones** entre los individuos. La elección de amigos que comparten los mismos puntos de vista es el factor que más reduce la exposición a opiniones discordantes, seguido por el sesgo que existe a la hora de elegir una noticia y que está dominado por la inclinación del usuario a buscar pruebas que **refuercen** sus puntos de vista. El algoritmo de ranking de Facebook también contribuye, pero en mucha menor medida.

El problema de la difusión de información en entornos sociales no es exclusivo de las redes sociales online, sino que ya existía antes de que éstas aparecieran. En un estudio publicado en el año 1995 [11], se muestra cómo en el mundo no virtual la pertenencia

a grupos cohesionados influye en la diversidad de **información política** a la que los individuos están expuestos.

En un estudio del 2014 [12], Colleoni, Rozza y Arvidsson utilizan técnicas de aprendizaje automático para **predecir** la orientación política de los usuarios de Twitter en base a sus publicaciones, y analizar el grado de **asortatividad** (la preferencia de los nodos de una red por unirse a otros que le son similares en alguna característica) de la red. Utilizan un enfoque similar al ya explorado por Pennacchiotti & Popescu [13] entrenando un **clasificador supervisado** Agresivo-Pasivo [14] para identificar la orientación política de los usuarios en base a los textos que publican, para posteriormente analizar los niveles de asortatividad en la red. Los resultados sugieren que el nivel de asortatividad difiere entre distintos grupos ideológicos, y depende en gran medida del uso que le dan los usuarios a Twitter.

2.2.2 Filtros burbuja en sistemas de recomendación y buscadores

Junto con las redes sociales, los buscadores se han convertido en una herramienta habitual en la distribución de noticias. Un **filtro burbuja** es el resultado de las **búsquedas personalizadas**, que pueden llevar al aislamiento del usuario en burbujas ideológicas y culturales propias. Los buscadores modernos utilizan la información que almacenan sobre los usuarios para ofrecer resultados **adaptados** a sus intereses, lo que puede resultar en una menor exposición a información contraria a sus puntos de vista.

Existe un debate acerca de si los buscadores realmente contribuyen a incrementar la segregación ideológica, o si por el contrario aumentan la exposición a perspectivas contrarias.

Los filtros burbuja y sus efectos en la opinión pública han sido estudiados a lo largo de los últimos años por varios autores [15][16]. El libro «**The filter bubble**» [16] explica los mecanismos que tienen los buscadores para adaptarse al usuario, y expone preocupación por el hecho de que los usuarios no sean del todo **conscientes** de que los resultados de un buscador varían para cada usuario, y por tanto no forman una imagen global sino parcial.

Los sistemas de recomendación son usados en multitud de servicios actualmente. Un **sistema de recomendación** intenta predecir qué temas o ítems serán del interés de un usuario, incluso antes de que éste los pruebe. Para ello, los algoritmos de recomendación más empleados utilizan los gustos e intereses de los usuarios que son más similares a dicho usuario para hacer predicciones acerca de sus gustos e intereses. Los buscadores como Google, Bing, etc pueden considerarse sistemas de recomendación de enlaces.

Un caso especial de sistemas de recomendación son los algoritmos de redes sociales como Facebook, Twitter, etc para **ordenar los contenidos**.

Los efectos del uso de sistemas de recomendación en la diversidad de contenidos a la que un individuo está expuesto han sido estudiados por un grupo de la universidad de Minnesota. En el estudio publicado en 2014 analizan un *dataset* del sistema de recomendación de películas MovieLens, y concluyen que los contenidos recomendados por el sistema se vuelven cada vez más restringidos y específicos a lo largo del tiempo.

Volviendo a los buscadores, otros estudios han investigado el efecto que tienen en áreas como la política y la **difusión de noticias**.

En un estudio del año 2006 [17], Introna y Nissenbaum plantean que los buscadores ejercen una gran influencia política por la capacidad que tienen de ocultar sistemáticamente ciertas páginas web (por diseño o accidentalmente) y favorecer otras.

El estudio realizado por Flaxman, Goel y Rao en 2016 [18] es más escéptico acerca de la influencia real de los buscadores. Defienden que el uso de redes sociales y buscadores en realidad no disminuye la exposición a información ideológicamente contraria, sino que la aumenta. Si bien admiten que el uso de éstas contribuyen a la **polarización**, aumentando la distancia ideológica media entre los individuos. Para conciliar ambos resultados defienden que la influencia de los buscadores en la difusión de noticias es mínima, ya que los accesos a noticias se realizan fundamentalmente a partir de las páginas de portada de los periódicos online, lo que contrasta con lo observado por otros autores, que defienden un papel importante de las redes sociales como medio de difusión.

2.3 Herramientas de recomendación de noticias

Actualmente existen un gran número de herramientas de recomendación de noticias. Algunas, como **Google News**, o **Bing News** surgen como una extensión de los buscadores. Otras se basan en el concepto de red social y son los propios usuarios los que recomiendan las noticias. Un ejemplo de red social orientada a noticias es **Menéame**, o las más generales **Twitter** y **Facebook**. Por último, existen los **agregadores de noticias** que utilizan el protocolo RSS, y que permiten suscribirse a un número de fuentes de noticias como periódicos o blogs y recibir actualizaciones. Muchos de estos agregadores, como **Feedly** o **Flipboard**, están incorporando sistemas que recomiendan suscripciones basándose en los contenidos suscritos por usuarios similares.

En esta sección se analizan algunas de las herramientas de recomendación de noticias más importantes.

2.3.1 Google News

Google News (`news.google.com`) es un agregador y buscador de noticias, que rastrea automáticamente los principales medios de comunicación online. Fue lanzado en enero de 2006, y es parte de la plataforma de Google Inc.

Utiliza un algoritmo de recomendación de noticias basado en el **filtrado colaborativo**, con el objetivo de ofrecer resultados personalizados a cada usuario [19]. El algoritmo evalúa la frecuencia y los sitios en los que es compartida una noticia, de forma que la promoción de noticias sea lo más neutral posible, al no incorporar intervención humana. Los resultados son personalizados para cada usuario en base al historial de búsquedas del propio usuario y del resto de usuarios, mediante un sistema de filtro colaborativo similar al usado por Amazon.

Actualmente el servicio no se encuentra accesible desde España, debido a la aplicación de la Ley de Propiedad Intelectual [20] [21].

2.3.2 Menéame

Menéame (`meneame.net`) es una red social de noticias española. Nació en el año 2005 de la mano de Ricardo Galli y Benjamí Villoslada. Menéame combina marcadores sociales, el blogging y la sindicación con un sistema de publicación sin editores. Desde sus inicios, el software fue liberado bajo la licencia Affero GPL.

A diferencia de Google News y otros agregadores de noticias, en Menéame son los propios usuarios quienes **proponen y votan** las noticias, decidiendo así el ranking de las noticias en la portada. Es este funcionamiento como red social el que permite que siga en funcionamiento en España a pesar de la Ley de Propiedad Intelectual [22].

Todos los datos almacenados, las estadísticas de los usuarios y los algoritmos utilizados son públicos, para garantizar la ausencia de manipulación en el

posicionamiento de noticias. Sin embargo, existen algunas críticas que afirman que el sistema de promoción no cumple con su misión porque las noticias más votadas son las más sensacionalistas.

2.3.3 Flipboard

Flipboard es un agregador de noticias, disponible para sistemas móviles y de escritorio. Permite agregar noticias de múltiples fuentes, incluyendo feeds RSS, blogs y redes sociales. Fue lanzado en 2010, originalmente para el dispositivo iPad de Apple, y dos años más tarde estuvo disponible para dispositivos Android.

El sistema de suscripción funciona mediante selecciones de contenido agrupadas en "magazines". Los usuarios pueden crear sus propios "magazines" seleccionando los temas que les interesan, o explorar las **recomendaciones** que les proporciona la aplicación.

2.4 Herramientas de análisis de redes sociales

En esta sección se exponen algunas de las herramientas y librerías más utilizadas, en el contexto del análisis de redes sociales.

2.4.1 Gephi

Gephi es una herramienta de visualización y exploración de grafos. Está programada en java, tiene licencia de software libre y es gratuita. Permite manipular grafos de gran tamaño mediante una interfaz gráfica. Está disponible para Windows, Mac OS X y Linux.

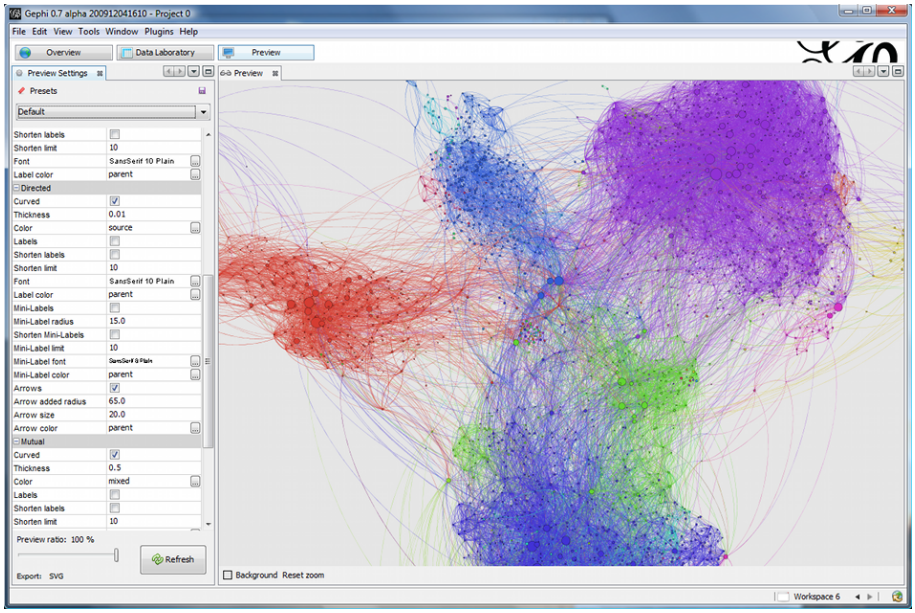

Figura 2.4: Ejemplo de uso de la herramienta Gephi para análisis y visualización de grafos

Como principal ventaja se encuentra el poder manejar grafos de gran tamaño (> 200.000 nodos), así como la disponibilidad de un gran número de algoritmos ya preparados. Esto lo hace una buena herramienta para una primera toma de contacto con un grafo.

2.4.2 NetworkX

NetworkX (https://networkx.github.io/) es un paquete de python para la creación y análisis de grafos. Implementa un gran número de algoritmos de análisis, y está liberado bajo licencia BSD-new.

Permite **exportar** e **importar** grafos en múltiples formatos, entre ellos GEXF, que es el utilizado por defecto en Gephi. Además, ofrece herramientas básicas de visualización basadas en la tradicional librería Matplotlib.

El hecho de estar disponible como una librería de **python** lo hace fácil de usar y de integrar con otras tecnologías.

```
2  import networkx as nx
3  import pylab as plt
4
5  G = nx.Graph() # crear grafo
6  G.add_edge('A','B', weight=1) # añadir enlaces con pesos
7  G.add_edge('A','C', weight=2)
8  G.add_edge('C','B', weight=2)
9  G.add_edge('A','D', weight=1)
10
11 print(nx.degree(G)) # calcular el grado de los nodos
12
13 nx.write_gexf(G, "test.gexf") # exportar el grafo en formato GEXF
14 nx.draw(G) # dibujar el grafo con matplotlib
15 plt.show()
```

Figura 2.5: Fragmento de código para crear un grafo con la librería NetworkX de python

2.4.3 Tweepy

Tweepy es una librería de python para acceder a la API de Twitter. Es de código abierto y está en activo desarrollo, lo que es indispensable para mantener compatibilidad con las APIs. Permite interactuar tanto con la **API REST** como con la **API Streaming**.

```
2  from tweepy.streaming import StreamListener
3  from tweepy import OAuthHandler, Stream
4
5  class listener(StreamListener):
6      def on_data(self, data):
7          print(data)
8          return True
9
10 auth = OAuthHandler(consumer_key, consumer_secret)
11 auth.set_access_token(access_token, access_token_secret)
12
13 stream = Stream(auth, listener())
14 stream.filter(track=['basketball', 'football'])
```

Figura 2.6: Fragmento de código para recolectar tweets en streaming con la librería Tweepy de python

2.4.4 Bases de datos NoSQL. MongoDB

Las bases de datos NoSQL permiten utilizar un modelo de datos más **flexible** que las bases de datos relacionales (RDBMS). Ésto es útil para almacenar datos **no estructurados**, como son generalmente los datos que provienen de redes sociales.

La mayor desventaja de las bases de datos NoSQL es la velocidad a la hora de realizar búsquedas complejas, debido al carácter no estructurado de los datos. Ésto puede solventarse mediante el uso de **índices**, siempre que se conozca de antemano el tipo de búsquedas que se van a realizar.

Un tipo de bases de datos NoSQL son las **orientadas a documentos**. Los documentos son paquetes de información semi estructurada, codificada en un cierto formato (XML, JSON, BSON, etc). Por ejemplo, un documento puede ser un tweet, en formato JSON.

MongoDB es una base de datos NoSQL orientada a documentos, que guarda estructuras de datos en formato BSON, similar a JSON. Permite realizar búsquedas por campo, creación de índices, replicación y balanceo de carga. Es adecuada para guardar *datasets* de gran tamaño. Sin embargo, no implementa las propiedades ACID en las transacciones, lo que la hace inadecuada para almacenar información sensible.

2.4.5 D3.js

D3.js (Data-Driven Documents) es una librería de javascript para producir visualizaciones dinámicas e interactivas a partir de datos. Permite tener un alto control sobre cómo son representados los datos.

Uno de los tipos de visualizaciones que permite crear son las visualizaciones de grafos, utilizando el algoritmo **force-directed layout**. Éste se basa en simular fuerzas atractivas y repulsivas con el objetivo de posicionar los nodos del grafo en un espacio de dos dimensiones, de forma que exista el menor número de cruces entre enlaces. Todos los nodos provocan fuerzas de repulsión entre sí, mientras que los enlaces provocan fuerzas atractivas que se oponen a las primeras, y que «acercan» los nodos en base al peso del enlace.

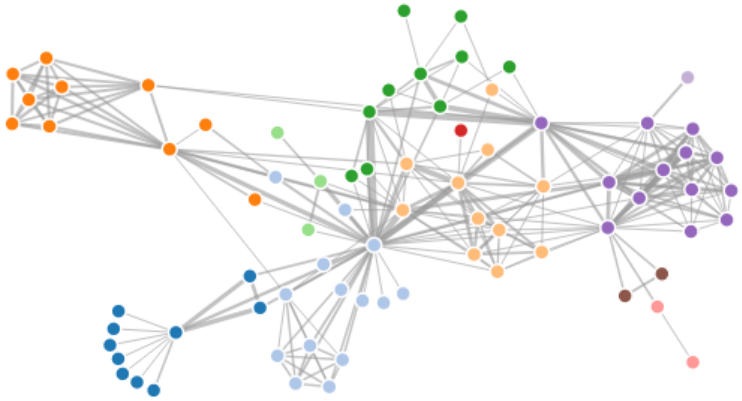

Figura 2.7: D3.js force layout

2.5 Algoritmos de detección de comunidades en grafos

Los grafos que representan sistemas reales no presentan características regulares y homogéneas. Por ejemplo, el número de enlaces por nodo (grado) suele seguir una **distribución exponencial**, y no homogénea como podría esperarse de un grafo aleatorio. No solo eso, sino que la distribución de enlaces tampoco es homogénea a nivel local, presentando una gran densidad de enlaces entre ciertos grupos de vértices, y menores enlaces entre dichos grupos. A esto se le llama **estructura de comunidades**. Los primeros algoritmos de detección de comunidades se basaban en un enfoque jerárquico, en el que los nodos se iban uniendo en función de una medida de prioridad. Posteriormente **Girvan y Newman** introdujeron en 2002 el que es uno de los algoritmos más importantes, basándose en el concepto de **intermediación** (betweenness) [23]. Desde entonces se han ido desarrollando otros algoritmos basados en ésta y otras medidas.

En esta sección se describen los algoritmos de detección de comunidades más importantes [5]:

2.5.1 Girvan Newman

La **intermediación** («betweenness centrality») es una medida que cuantifica la frecuencia o el número de veces que un nodo actúa como un puente a lo largo del camino más corto entre otros dos nodos. Para cada par de nodos en el grafo, el camino más corto entre ellos es aquél que minimiza el número de enlaces (o la suma de los pesos de éstos). La intermediación de un nodo o de un enlace define como:

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}(v)}$$

Donde σ_{st} es el número de caminos más cortos entre los nodos s y t, y $\sigma_{st}(v)$ es el número de esos caminos que pasan por el nodo/enlace v.

Si hay dos o más caminos cortos alternativos, con igual número mínimo de enlaces, a cada camino se le asigna una **fracción** del peso, sumando en total 1. La idea central del algoritmo es la siguiente: si existen dos comunidades conectadas con un número reducido de enlaces inter-comunidad, dichos enlaces tendrán una alta intermediación, ya que todos los caminos entre nodos de comunidades separadas deberán pasar por ellos.

El **algoritmo de Girvan Newman** procede de la siguiente forma [23]:

1. Calcular la intermediación para todos los enlaces de la red.
2. Eliminar el enlace con la intermediación más alta.
3. Recalcular la intermediación para todos los enlaces afectados por el paso anterior.
4. Repetir desde el segundo punto hasta eliminar todos los enlaces.

El resultado es una **jerarquía de particiones** del grafo en comunidades. Para seleccionar la mejor partición existen varias métricas posibles, siendo la **modularidad** la más utilizada.

2.5.2 Louvain Method

El método **Louvain** fue publicado en 2008 como un método para extraer la estructura de comunidades en grafos de tamaño elevado (del orden de millones de nodos y billones de enlaces). Está basado en heurísticas orientadas a optimizar la modularidad.

El algoritmo está dividido en dos fases, que se repiten indefinidamente:

1. Inicialmente, a cada nodo se le asigna su propia comunidad, habiendo tantas comunidades como nodos. Posteriormente, para cada nodo i se consideran sus nodos vecinos j, y se considera la **ganancia de modularidad** que se obtendría colocando i en la comunidad de j. Esto se realiza para cada nodo vecino j, y finalmente se coloca i en la comunidad que proporcione la ganancia máxima de modularidad. Este proceso es repetido para todos los nodos hasta que no se pueda obtener ninguna ganancia, es decir, cuando se llega a un máximo local de la modularidad.

2. La segunda fase del algoritmo consiste en la creación de un nuevo grafo en el que los nodos son las comunidades encontradas en el paso anterior.

La siguiente figura muestra un ejemplo de este proceso:

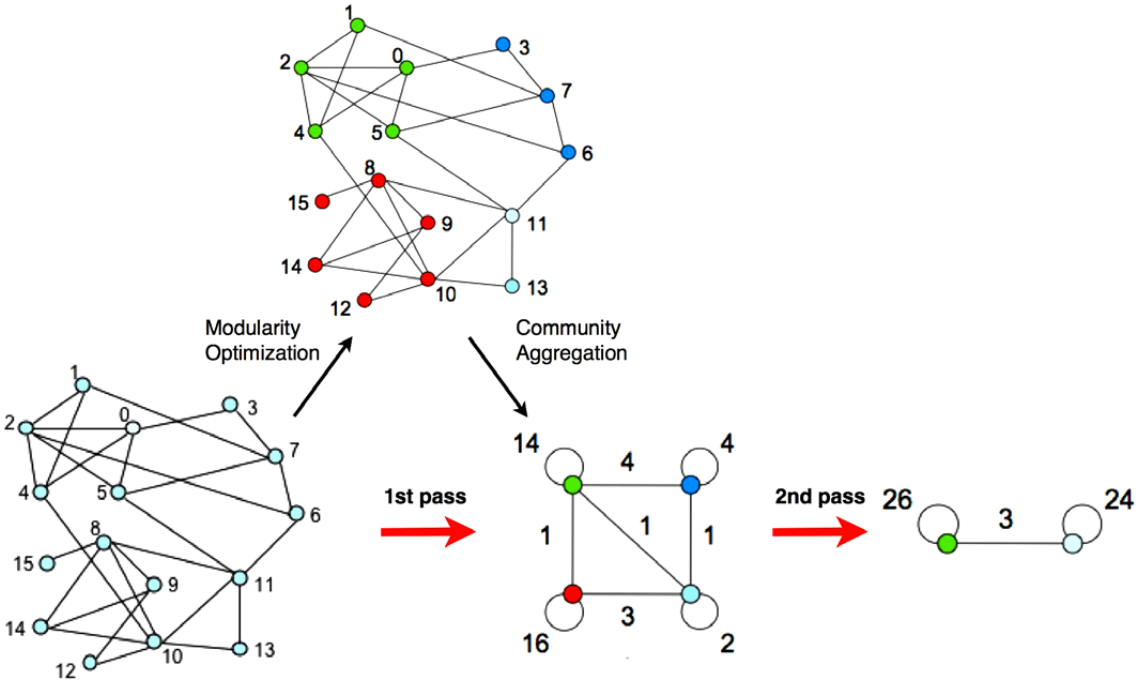

Figura 2.8: Fases del método Louvain. *Fast unfolding of communities in large networks* [3]

2.5.3 Label Propagation

El algoritmo **Label Propagation**, o propagación de etiquetas fue publicado en 2007 [24]. Cada nodo tiene asociada una **etiqueta**, que representa la comunidad a la que pertenece. En cada paso del algoritmo se elige un nodo. Para ese nodo, se observan sus nodos vecinos, y se selecciona la etiqueta mayoritaria entre ellos. El nodo pasa entonces a formar parte de esa comunidad.

Conforme las etiquetas se **propagan**, los grupos de nodos conectados densamente convergen rápidamente a una única etiqueta, y formarán cada una de las comunidades.

Este proceso puede realizarse de manera **síncrona** o **asíncrona**. En la versión síncrona, cada nodo elige su etiqueta basándose en las etiquetas de sus vecinos en el paso anterior. Esto puede conllevar problemas de convergencia debido a oscilaciones continuadas. La versión asíncrona soluciona este problema, siendo cada nodo actualizado en orden, utilizando las etiquetas actuales de sus vecinos. El orden de actualización es elegido de manera aleatoria.

2.5.4 Fluid Communities algorithm

El algoritmo **Fluid Communities** [4] se basa en la idea de fluidos interactuando en la naturaleza, expandiéndose y contrayéndose como resultado de la interacción. Cada fluido representa una comunidad, y el algoritmo es capaz de detectar un número predeterminado de comunidades.

Inicialmente, se elige un número k de comunidades, y se inicializa cada comunidad en un nodo aleatorio del grafo. Cada comunidad tiene asociada una densidad d, que toma valores en el rango $(0, 1]$.

En cada iteración, el algoritmo itera sobre un nodo, actualizando su comunidad en función de las de sus vecinos. Para elegir, se tiene en cuenta el número de vecinos que corresponden a cada comunidad, pero también las densidades de cada comunidad. Así, cada nodo aporta d a la suma, donde d se corresponde con la densidad de la comunidad a la que pertenece. Se elige la comunidad con **mayor densidad agregada**.

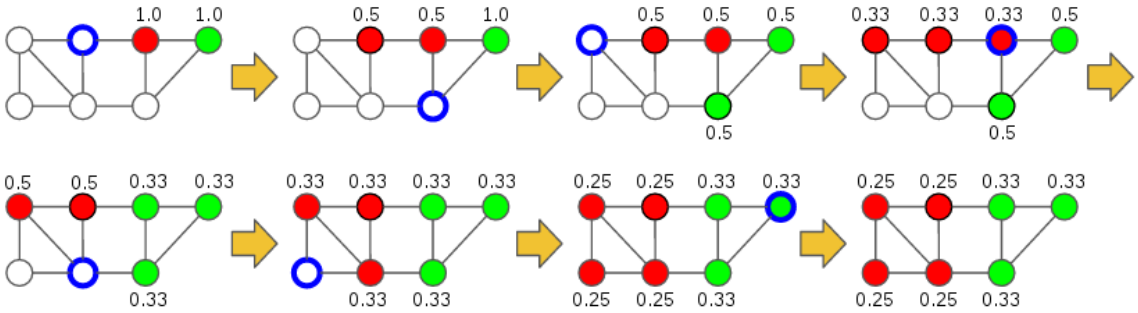

Figura 2.9: Para cada vértice se muestra la densidad de la comunidad a la que pertenece. *Fluid Communities: A Competitive, Scalable and Diverse Community Detection Algorithm* [4]

3 SISTEMA DESARROLLADO

3.1 Esquema funcional

A continuación se muestra un **esquema funcional** del proyecto, donde se puede ver cómo interaccionan los distintos componentes.

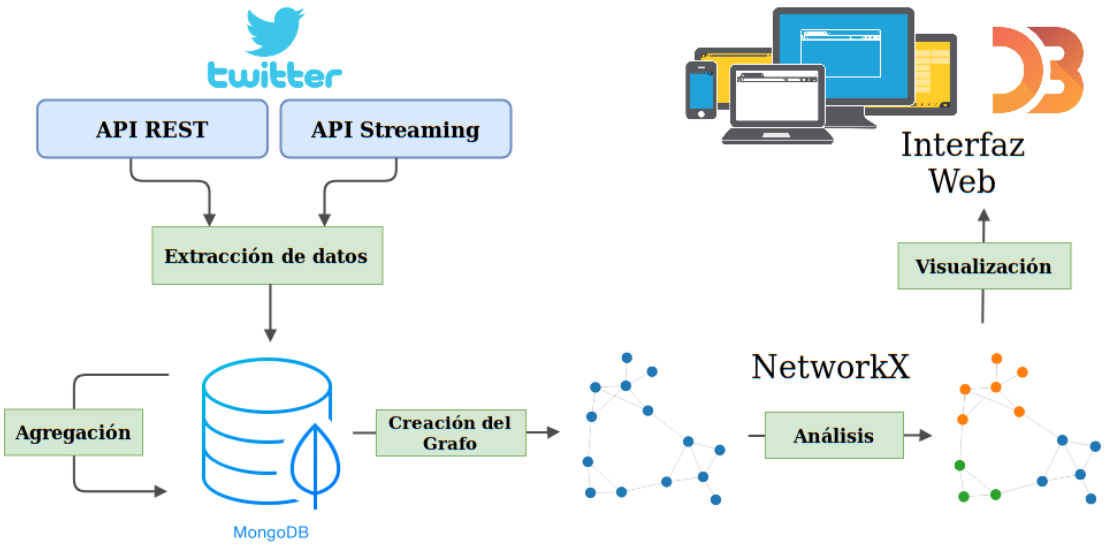

Figura 3.1: Esquema funcional

En verde se muestran las cinco etapas por las que pasa el **flujo de datos** desde que son recolectados hasta que son mostrados al usuario:

- **Extracción de datos:** en este capítulo se hablará de la conexión con las APIs de twitter y de la base de datos utilizada.
- **Agregación:** en este capítulo se hablará de los procesos de agregación, actualización y enriquecimiento que se aplican a los tweets.
- **Creación del grafo:** en este capítulo se explica la forma en la que se transforman los datos de tweets en el grafo de noticias.

- **Análisis:** detalla los algoritmos y técnicas de análisis de grafos utilizados. En concreto, las técnicas para simplificar el grafo y los algoritmos de detección de comunidades.
- **Visualización:** en este capítulo se hablará de la herramienta construida para visualizar el grafo y cómo usarla para analizar el grafo.

3.2 Extracción de datos

Los datos utilizados en este proyecto han sido extraídos de la **API pública de Twitter**. Se descartó la opción de otras redes sociales como Facebook e Instagram, debido a las limitaciones de búsqueda y acceso de sus respectivas APIs, en concreto la imposibilidad de realizar búsquedas por texto o palabras clave.

3.2.1 API de Twitter

La plataforma de Twitter para desarrolladores se divide en dos APIs: la **API REST**, y la **API Streamming**. Ambas APIs tienen limitaciones, pero se pueden combinar para recolectar la mayor cantidad posible de información. También existe una API Enterprise de pago, que elimina todas las limitaciones.

La **API tipo REST** permite realizar búsquedas en el **histórico** de tweets. Sin embargo, tiene limitaciones importantes tanto en el número de peticiones por minuto, como en la cantidad de resultados. Se puede obtener más información acerca de las limitaciones de la API en el siguiente enlace: `https://developer.twitter.com/en/docs/basics/rate-limits`

La **API tipo Streamming** permite recolectar tweets que están siendo publicados en **tiempo real**, proporcionando una mayor cantidad de tweets que la API REST.

Para conectar con las APIs se ha utilizado python, con la librería **tweepy**. Se ha utilizado la base de datos **MongoDB** junto con la librería **pymongo** para guardar los tweets recolectados.

3.2.2 Servidor

Para realizar la extracción de datos se ha contratado un **servidor remoto**.

El servidor mantiene tres **servicios** activos: los scripts de recolección de tweets y *likes*, la base de datos MongoDB, además del servidor SSH.

```
javi@ubuntu:~# netstat -ntpl
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:55622           0.0.0.0:*               LISTEN      746/sshd
tcp        0      0 0.0.0.0:27017           0.0.0.0:*               LISTEN      1718/mongod
tcp6       0      0 :::55622                :::*                    LISTEN      746/sshd
javi@ubuntu:~# supervisorctl
twitter_streamming               RUNNING   pid 2524, uptime 2 days, 4:46:44
update_likes                     RUNNING   pid 11884, uptime 0:22:53
supervisor>
```

Figura 3.2: Servicios corriendo en el servidor

Para mantener activos los scripts se hace uso de **supervisord**, un programa que permite monitorizar y controlar la ejecución de procesos en sistemas **UNIX/Linux**.

Los dos scripts se mantienen recolectando datos las 24 horas. El primero **recolecta** los tweets en streaming y los guarda en MongoDB. El segundo va **actualizando** los *likes* de

cada tweet periódicamente, guardando un *timestamp* para cada uno de ellos. En total se han recolectado casi 5 millones de tweets.

```
MongoDB server version: 3.6.4
>
> use tfg
switched to db tfg
>
> db.getCollection('tweets_with_urls').find({}).count()
4873548
>
> db.getCollection('tweets_with_urls').findOne({})
{
        "_id" : ObjectId("5a1b038179aed06baf82ccc5"),
        "tweet" : {
                "created_at" : "Sun Nov 26 18:10:09 +0000 2017",
                "id" : NumberLong("934846755014889472"),
```

Figura 3.3: Conexión con la base de datos MongoDB

3.2.3 Extracción

La extracción de datos se lleva a cabo en dos fases:

En la **primera fase** se recolectan tweets relacionados con el tema de interés desde la API REST y la API Streaming. Si es un **suceso pasado**, la única fuente de tweets será la API REST, que permite el acceso a tweets históricos. Si por el contrario es un suceso que sabemos que ocurrirá en el **futuro inmediato** (ej. unas elecciones), se utiliza la API Streaming, que permite recolectar una mayor cantidad de tweets. Como trabajo futuro se plantea incorporar algoritmos de detección automática de sucesos en tiempo real, para poder utilizar la API Streamming con antelación y disponer de un mayor número de tweets en todos los casos [25][26].

Únicamente se guardan los tweets que contienen urls enlazadas, el resto son descartados.

La **segunda fase** consiste en **recolectar** y **actualizar** los "*likes*" de cada tweet, esto es, las ids de los usuarios que han dado like a cada tweet. Para ello se hace uso de un script que se ejecuta periódicamente, conecta con la API REST, y va actualizando los registros en MongoDB. Los tweets recolectados en streaming no contienen inicialmente ningún like, por lo que este paso es de especial importancia. El script que realiza la actualización se ejecuta dos veces al día. Es necesario esperar un tiempo desde que el tweet se publica hasta que aparecen suficientes *likes*, al menos 24 horas.

3.3 Agregación

El siguiente paso consiste en **agrupar** todos los tweets que publican la **misma url**. El uso de **acortadores** de url dificulta esta tarea. Para solventarlo, se ha hecho uso del servicio **unshorten.me**, que proporciona una API gratuita para la resolución de urls acortadas.

Una vez acortadas, se almacenan las urls junto con los tweets asociados. Esto permitirá posteriormente la creación del grafo.

Además, para cada noticia almacenada se ha extraído **información** que será usada tanto en esta etapa como posteriormente en la etapa de visualización. Para ello se ha usado el protocolo **Open Graph Protocol** (`http://ogp.me/`). OPG es un protocolo creado por Facebook en 2010, y que es implementado por la mayoría de periódicos online en sus páginas de noticias. Permite enriquecer cualquier página web mediante el uso de ciertos tags que se insertan en el código html. Orientado al uso en Facebook, convierte la página

en un nodo de la red social, definiendo la información que aparece cuando la página es compartida.

```
<html prefix="og: http://ogp.me/ns#">
<head>
<title>The Rock (1996)</title>
<meta property="og:title" content="The Rock" />
<meta property="og:type" content="video.movie" />
<meta property="og:url" content="http://www.imdb.com/title/tt0117500/" />
<meta property="og:image" content="http://ia.media-imdb.com/images/rock.jpg" />
...
</head>
...
</html>
```

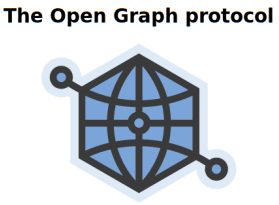

Figura 3.4: Ejemplo de tags usados en OpenGraph

Los **metadatos** permiten obtener el título, un resumen, y la imágen de portada de la noticia. Esta información será útil para mejorar la **visualización** y la comprensión del grafo,

Se utiliza el título y el resumen para **descartar noticias** que no tienen relación con el suceso buscado(no contienen ninguna de las palabras clave), y que por alguna razón han llegado hasta este paso. También permiten determinar si dos urls son en realidad la misma noticia.

3.4 Creación del grafo

La **creación del grafo** se realiza a partir de los datos almacenados en MongoDB y utilizando la librería **NetworkX** de python.

Consideramos que un usuario **está de acuerdo** con una noticia si da "like" a un tweet que publica dicha noticia.

Cada noticia conforma un nodo del grafo. Para determinar los enlaces entre noticias, usamos los *likes* de los usuarios del siguiente modo: si un usuario da like a **dos noticias distintas**, manifiesta que está de acuerdo con ambas, y por tanto las "enlaza".

Cada url tiene una lista con los usuarios que le han dado like o que la han tuiteado. Se define la "proximidad" entre un par de noticias como el tamaño de la **intersección** entre ambas listas de usuarios.

```
62
63  ## Crear Grafo
64  G=nx.Graph()
65  pairs = itertools.combinations(urls, 2)
66  for url1, url2 in pairs:
67      users_in_common = set(url1['likes']).intersection(set(url2['likes']))
68      prox = len(users_in_common)
69      if prox >= MIN_PROX: # proximidad mínima para considerar un enlace
70          G.add_edge(url1['url'], url2['url'], weight=prox)
71
72
```

Figura 3.5: Creación del grafo. El peso de cada enlace corresponde al número de usuarios que dieron like a ambas noticias.

El resultado es un **grafo no dirigido** y con **pesos**, en el que cada enlace tiene como peso el número de usuarios que dan like a ambas noticias. Dos noticias con un punto de vista cercano, estarán probablemente **enlazadas** con un peso alto (es probable que muchos usuarios estén de acuerdo con ambas), mientras que dos noticias enfrentadas

aparecerán **lejanas** en el grafo (pocos o ningún usuario está de acuerdo con ambas al mismo tiempo).

3.5 Análisis del grafo

El análisis del grafo consta de dos fases: En la **primera**, se simplifica el grafo para reducir el número de enlaces manteniendo las estructuras más importantes. En la **segunda**, se utilizan algoritmos de detección de comunidades.

3.5.1 Simplificación

La **densidad** de enlaces del grafo depende en gran medida del número de tweets recolectados y del impacto del suceso en las redes sociales. Para que dos noticias estén relacionadas basta con que un usuario de like a ambas o las retweetee. Esto puede generar un grafo **muy densamente** conectado, que dificulta el análisis y la visualización posterior. Para solventar esto, se utilizan dos estrategias de reducción del número de enlaces.

La primera estrategia consiste en eliminar durante la creación del grafo aquellos **usuarios** que dan un número excesivo de *likes*. En la siguiente figura se observa que el número de *likes* que proporcionan los usuarios es exponencial, con algunos usuarios proporcionando un número muy elevado, de forma poco discriminada, mientras que la mayoría de usuarios únicamente proporcionan uno o dos *likes*. Consideramos que los usuarios que proporcionan menor número de *likes* son más selectivos y por tanto más fiables.

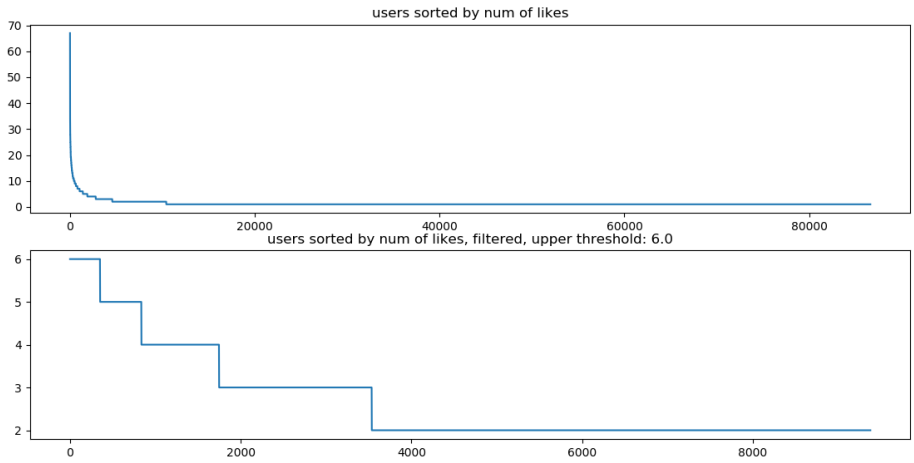

Figura 3.6: Arriba, los usuarios ordenados por número de noticias a las que dan like, sobre un total de 500 noticias. Abajo, los mismos usuarios en este caso filtrando aquéllos que proporcionan un exceso de *likes*, y los que proporcionan únicamente uno.

El número de *likes* que se considera excesivo dependerá del suceso analizado, y del número total de noticias. Por ello, se utiliza un test estadístico (**test de Turkey**) para detectar **valores atípicos**(outliers) [27]. Los valores atípicos son aquéllos que superen un umbral q, definido como:

$$q > Q_3 + 3 * IQR$$

Donde Q_3 determina el tercer cuartil, e IQR es el rango intercuartil.

La segunda estrategia consiste en **podar** enlaces del grafo, para reducir la densidad global. Debido a la naturaleza del grafo, existen comunidades con tamaño y densidad muy altas, que coexisten con comunidades más pequeñas (estas comunidades pequeñas pueden ser de especial interés ya que pueden representar grupos de opinión minoritarios). La estrategia de reducir un número fijo de enlaces por nodo, o la de seleccionar enlaces de manera aleatoria no funcionan ya que las comunidades más pequeñas **desaparecen** pronto. Por ello, se utiliza una estrategia de poda basada en la **densidad** de enlaces, que permite preservar la estructura de comunidades.

La estrategia de poda es la siguiente:

- Para cada nodo se mantiene un número de enlaces variable. Se reduce a la mitad el número de enlaces, manteniendo un mínimo de 1.
- Los enlaces a eliminar son seleccionados mediante una función de puntuación que tiene en cuenta el peso del enlace y la centralidad del nodo con el que conecta. La **centralidad** de un nodo está definida como la fracción de los nodos del grafo con los que está conectado [28].
- Debido a que al eliminar un enlace éste afecta a dos nodos, es conveniente empezar la poda por los nodos menos centrales. De esta forma se asegura que los nodos menos centrales quedan conectados a los nodos más centrales, que podrían considerarse los **núcleos de las comunidades**.

3.5.2 Detección de comunidades

Para el análisis del grafo se han utilizado algoritmos ya existentes implementados en la librería **NetworkX** para python. Se ha realizado un trabajo de adaptación de dichos algoritmos al problema concreto, a las características concretas del grafo, y a los objetivos deseados.

El objetivo del análisis del grafo es encontrar una lista de comunidades que representen los puntos de vista más diferenciados de todo el conjunto de noticias. Los algoritmos utilizados deben poder encontrar **comunidades** de distintos tamaños y densidades, así como poder trabajar con grafos con pesos.

Se ha realizado una búsqueda en la literatura existente [5][29][3][23], y se ha procedido a seleccionar aquellos algoritmos que se adaptan a los criterios mencionados anteriormente, y para los que existe una implementación en python.

Los algoritmos seleccionados son los siguientes:

- Girvan Newman
- Louvain (Modularity)
- Label Propagation
- Fluid communities

En la sección 2.5 se da una introducción a estos algoritmos.

3.6 Visualización. Interfaz web

El objetivo último de la herramienta es poder ser usada por usuarios finales con nociones básicas de grafos, pero sin conocimientos de programación. Por ello, se ha desarrollado una interfaz gráfica que permita explorar los resultados de una manera interactiva y accesible.

Se ha elegido desarrollarla usando **tecnologías web**, debido a que los usuarios están muy habituados a su uso, y para facilitar la integración con otros servicios.

La interfaz web ha sido escrita en **javascript**, usando **D3.js** como librería gráfica, utilizando el algoritmo Force Directed Layout para colocar los nodos. En el lado del servidor se utiliza el framework **Flask**, escrito en python. Flask es un framework minimalista con licencia de código abierto.

La comunicación entre ambas partes (**frontend y backend**) se realiza usando **AJAX**. El grafo es transmitido del servidor al cliente en formato **JSON**.

La siguiente figura muestra un ejemplo de la interfaz web:

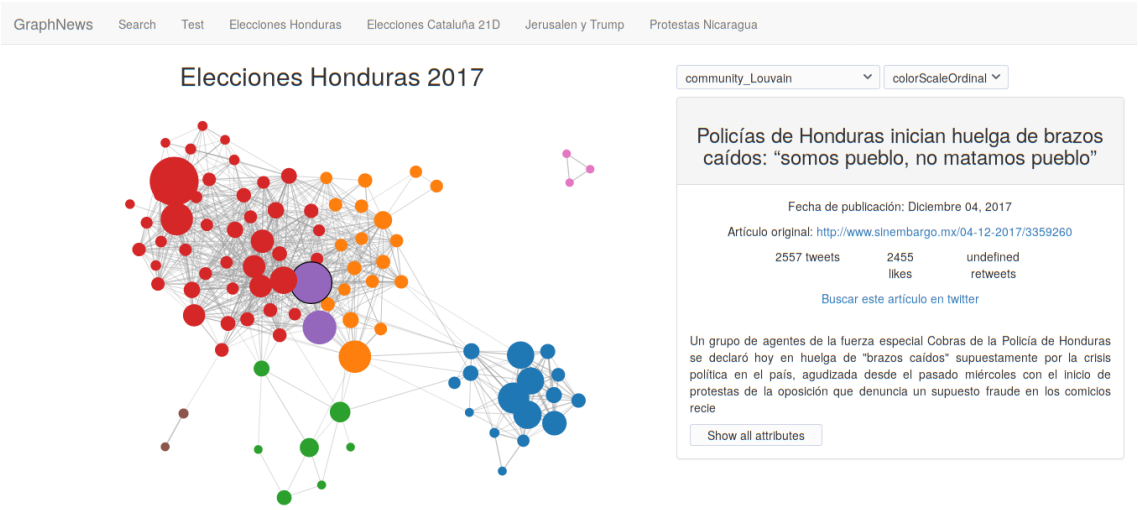

Figura 3.7: Interfaz web creada para analizar el grafo de noticias

El **tamaño** de los nodos muestra el número de **likes y retweets** de la noticia asociada, mientras que **color** de los nodos muestra el resultado del algoritmo de detección de **comunidades** seleccionado.

Al hacer click en cualquiera de los nodos, se muestra un recuadro de la derecha con la **información asociada** a dicha noticia, y un enlace a la publicación original.

El recuadro muestra la siguiente información:

- Título
- Fecha de publicación
- Resumen de la noticia
- Número de *likes*

- Número de veces que la noticia ha sido tweeteada ó retweeteada
- Enlace a la noticia original

En la siguiente figura puede verse cómo la interfaz permite cambiar el coloreado de los nodos seleccionando entre los distintos **algoritmos** de detección de comunidades. Esto permite comparar fácilmente el resultado de los distintos algoritmos:

Figura 3.8: La interfaz permite elegir uno de los múltiples algoritmos de detección de comunidades, para el coloreado de los nodos

4 Pruebas y Resultados

4.1 Objetivos

El **objetivo** de las pruebas es comprobar el correcto funcionamiento de la herramienta, analizando una lista de noticias reales, y comprobando que cumple los siguientes objetivos:

- Permite **recolectar y almacenar** los datos necesarios a partir de la API de Twitter
- Permite **transformar** los tweets en un grafo de noticias
- Permite aplicar algoritmos de detección de comunidades para **analizar** el grafo
- Permite al usuario **visualizar** y analizar los resultados de manera interactiva, e identificar los distintos grupos de opinión

La interfaz está pensada para usuarios con nociones básicas de los conceptos de grafos y comunidades. Durante las pruebas con usuarios no técnicos se hará una **introducción** previa para explicarles el funcionamiento de la herramienta y los datos representados.

Se han realizado **tres pruebas** con noticias reales. Las noticias han sido seleccionadas atendiendo a las siguientes características:

- Son noticias con **gran impacto** en las redes sociales y que por tanto han generado un gran número de tweets
- Han generado **diversidad de opiniones**, por lo que son idóneas para identificar grupos de opinión

Se ha estudiado la **calidad** de las comunidades encontradas mediante el uso de una métrica de calidad no supervisada, una métrica de rendimiento supervisada, y la evaluación visual por parte de usuarios mediante la interfaz gráfica.

Después se ha analizado la relación que se da entre el algoritmo que mejor identifica las comunidades desde el punto de vista del usuario, y el resultado de las distintas métricas.

4.2 Métricas

Se han utilizado dos tipos de métricas para evaluar los resultados de los algoritmos de detección de comunidades. El primer tipo consiste en medidas de calidad **no supervisadas**, que analizan únicamente características estructurales del grafo, sin atender al significado de los datos que representa. El segundo tipo de métricas utiliza un **conjunto de datos de prueba**, que ha sido creado a medida para cada experimento con ayuda de usuarios reales. Los datos de prueba consisten en pares de noticias para los que se determina si deben ir en comunidades distintas o en la misma.

4.2.1 Medidas de calidad supervisadas: Performance, coverage y modularidad

Una **partición** de un grafo es un conjunto de comunidades que cubren todos los nodos de éste.

Una **medida de calidad** se define como una función que asigna un número a cada partición de un grafo. Permite ordenar distintas particiones de un mismo grafo en función de la puntuación obtenida [5].

Las medidas de calidad se permiten evaluar la calidad de una partición sin atender al significado de lo que representan sus nodos y sus comunidades. Las más utilizadas son: **performance**, **coverage** y **modularity**.

La medida **performance** se define como aquella que penaliza el número de enlaces entre nodos de diferentes comunidades, a la vez que puntúa el número de enlaces intra-comunidad:

$$Performance = \frac{|\{(i,j) \in E, C_i = C_j\}| + |\{(i,j) \notin E, C_i \neq C_i\}|}{n(n-1)/2}$$

La medida **coverage** se define como el ratio de enlaces intra-comunidad entre el número de enlaces total.

La medida de calidad más utilizada es la **modularidad**. Ésta se basa en la idea de comparar la densidad de enlaces dentro de una comunidad con la que aparecería en un grafo aleatorio. El grafo aleatorio que se utiliza como modelo puede variar, pero debe contener el mismo número de enlaces que el grafo original. En este caso se ha utilizado un grafo aleatorio de Bernoulli. La modularidad se define como:

$$Modularidad = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(C_i, C_j)$$

- m es el número total de enlaces del grafo
- A es la matriz de adyacencia del grafo Pij es el número esperado de enlaces entre los nodos i y j en el modelo
- δ(Ci, Cj) vale 1 si los nodos i y j están en la misma comunidad, 0 en otro caso

4.2.2 Medidas de calidad supervisadas: Rand index, Ratio de verdaderos positivos y Ratio de verdaderos negativos

Las medidas de calidad supervisadas consisten en comparar la partición obtenida con una **partición modelo** previamente construida a partir de conocimiento experto.

En la literatura existen tres tipos de medidas para comparar particiones: basadas en *contar pares*, basadas en *emparejar clusters* y basadas en *teoría de la información*.

En este proyecto se usa el enfoque del **conteo de pares**, por ser el que menos suposiciones hace sobre la estructura correcta de las comunidades.

Con ayuda de un usuario se identifican manualmente pares de noticias que deben ir en comunidades separadas, o que deben ir en la misma comunidad. Posteriormente se comprueban en el grafo cuantos de estos pares han quedado correctamente separados o agrupados.

El **índice Rand** se define como el ratio entre el número de pares que han sido clasificados correctamente entre el total de pares [5]:

$$RandIndex = \frac{a_{11} + a_{00}}{a_{11} + a_{01} + a_{10} + a_{00}}$$

Donde:

- a_{11}: par de noticias correctamente unido en la misma comunidad (verdadero positivo)
- a_{01}: par de noticias incorrectamente unido en la misma comunidad (falso positivo)
- a_{10}: par de noticias incorrectamente separado (falso negativo)
- a_{00}: par de noticias correctamente separado (verdadero negativo)

Este valor coincide con la medida **accuracy** del error estadístico.

Otras medidas útiles son los valores **true negative rate** (también llamado specificity) y **true positive rate** (también llamado recall), que analizan por separado el número de noticias correctamente separadas y el número de noticias correctamente unidas:

$$TrueNegativeRate = \frac{TN}{TN + TP} \qquad TruePositiveRate = \frac{TP}{TN + TP}$$

Hay que tener en cuenta que la partición que maximiza el **true negative rate** es aquella que separa todos los nodos, asignando a cada uno su propia comunidad.

De igual forma, la partición que maximiza el **true positive rate** es aquella que asigna a todos los nodos la misma comunidad.

4.3 Metodología de pruebas

A continuación se explica en qué consistieron las distintas fases de las pruebas realizadas.

- **Selección de un suceso**: Se selecciona un suceso de relevancia, que esté generando un gran número de publicaciones en Twitter.
- **Recolección de datos**: Se configura la herramienta para recolectar los tweets durante un periodo de tiempo de al menos 24 horas.
- **Procesamiento de los datos y generación del grafo**: Se utiliza la herramienta construida para construir el grafo de noticias a partir de los datos recolectados.

- **Análisis del grafo junto a un usuario**: Se explica al usuario el funcionamiento de la herramienta, y se exploran los resultados de los distintos algoritmos. El usuario proporciona una valoración e identifica manualmente los principales grupos de opinión que observa en las noticias, y el algoritmo que mejor los identifica. Para cada grupo de opinión selecciona las noticias principales.
- **Análisis de los resultados de las métricas**: Se analiza el grafo con las distintas métricas y se contrastan los resultados de cada una con los proporcionados por el usuario.

Las valoraciones de los usuarios son recogidas en cada una de las pruebas, y son utilizadas para mejorar los resultados de las siguientes.

4.4 Pruebas realizadas

En esta sección se detallan las **pruebas** realizadas. Se han analizado sucesos reales, ocurridos a lo largo de los últimos meses de 2017 y primeros de 2018.

Las noticias han sido seleccionadas en base al volumen de publicaciones que han generado, y a la diversidad de opiniones que han producido.

4.4.1 Elecciones en Cataluña, 2017

Las últimas elecciones en Cataluña generaron una gran polarización en la sociedad, que se reflejó en las redes sociales.

Las **palabras clave** usadas para recolectar tweets fueron las siguientes: *21D, elecciones cataluña, eleccions, catalonia elections*. Se recopilaron alrededor de 1 millón de tweets que enlazaban noticias, publicados los días previos, durante, y posteriores a las elecciones.

La figura muestra el grafo generado con las 500 noticias más twiteadas, con las comunidades obtenidas con el algoritmo Girvan-Newman con *k=5*:

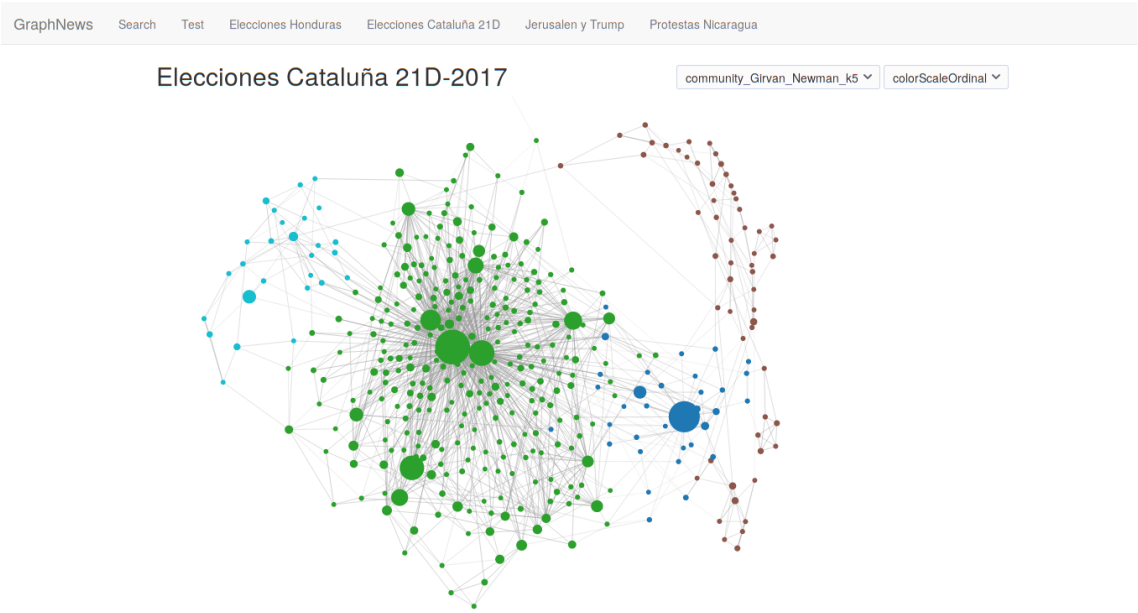

Figura 4.1: Elecciones catalanas. Se muestran las 500 noticias más twitteadas.

El algoritmo que mejores resultados proporciona es el de Girvan Newman, con *k=5*.

Comentarios tras las pruebas: El usuario manifestó las siguientes impresiones:

- El significado del grafo y sus relaciones no estaba clara al principio. Tras una explicación, el usuario entendió mejor la representación, y empezó a extraer significado.
- Llama la atención la considerable **diferencia** de tamaño entre las comunidades, con la comunidad verde conteniendo más de la mitad de las noticias publicadas
- Por otro lado, llama la atención el grado de conexión de la comunidad azul claro (de **noticias internacionales**), con la comunidad verde.
- Las comunidades representadas ayudan a la comprensión de la información. Las comunidades obtenidas con el algoritmo Girvan Newman coinciden con el **conocimiento previo** que el usuario tenía acerca del tema.
- El número de noticias mostradas es algo excesivo.

Durante el análisis el usuario identificó **cuatro comunidades** (las mostradas en la imágen), representando los siguientes grupos de opinión:

- Comunidad verde: noticias de periódicos a favor de la independencia
- Comunidad marrón: noticias de periódicos en contra de la independencia
- Comunidad azul claro: noticias de medios extranjeros, en inglés.
- Comunidad azul oscuro: medios con una posición poco definida

En la siguiente tabla se muestra la puntuación de las diferentes **métricas** para cada uno de los algoritmos. Se ha incluido un algoritmo que asigna comunidades de forma aleatoria:

Resultados de las métricas						
	Modularidad	Coverage	Performance	RandIndex	TrueNeg	TruePos
Louvain	0.319	0.519	0.893	0.365	0.954	0.156
GirvanNewman k=3	0.074	0.989	0.279	0.828	0.388	0.984
GirvanNewman k=4	0.098	0.971	0.380	0.900	0.710	0.967
GirvanNewman k=5	0.122	0.938	0.494	0.900	0.748	0.954
GirvanNewman k=6	0.131	0.926	0.517	0.842	0.759	0.872
Label Propagation	0.127	0.891	0.521	0.842	0.833	0.845
Fluid Comm k=4	0.200	0.658	0.675	0.513	0.811	0.407
Random k=4	-0.080	0.226	0.743	0.378	0.743	0.248

Tabla 4.1: Métricas de calidad. Elecciones Cataluña, 2017

Los algoritmos que mejor puntúan en el RandIndex son el de Girvan Newman con *k=5,4,6* seguido por Label Propagation. El algoritmo de Louvain puntúa alto en true negatives, y muy bajo en true positives lo que es señal de que el número de comunidades que crea es demasiado elevado.

4.4.2 Manifestaciones tras elecciones en Honduras, 2017

El 26 de noviembre de 2017 se celebraron en Honduras las elecciones para la presidencia del estado. Durante las semanas siguientes se produjeron manifestaciones acusando al gobierno de fraude electoral.

Las **palabras clave** que se usaron para recolectar tweets fueron las siguientes: *elecciones Honduras, Honduras*

Se recopilaron 900.000 tweets enlazando artículos y noticias.

Teniendo en cuenta las opiniones expuestas por el usuario en la etapa anterior al respecto del exceso de noticias, en este caso se seleccionaron las **100 noticias** más importantes en número de tweets.

El grafo generado se muestra en la siguiente figura:

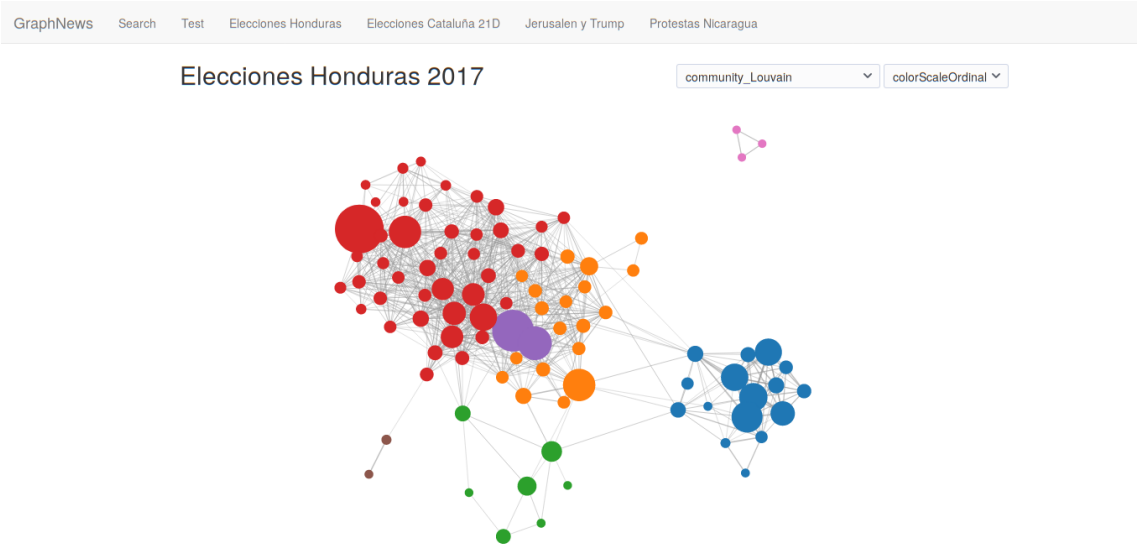

Figura 4.2: Manifestaciones tras las elecciones de 2017 en Honduras. Se muestran las 100 noticias más twitteadas.

Comentarios tras las pruebas: El usuario manifestó las siguientes impresiones:

- Los algoritmos detectan correctamente que existen **dos grupos de opinión enfrentados**, que corresponden con la **comunidad roja** y la **azul**. Esto coincide con lo esperado, debido a la polarización que existe en el país.
- Es interesante observar que la **comunidad azul** incluye comunicados oficiales de los países Perú y Argentina.
- Las otras comunidades detectadas son interesantes. Tras analizarlas, se observa que la **comunidad naranja** representa noticias de periódicos que no se posicionan en ningún extremo claro, incluyendo las noticias de **medios europeos y estadounidenses**; mientras que la **comunidad verde** corresponde enteramente a noticias publicadas por un periódico venezolano.
- El número de noticias mostradas no resulta excesivo.

En la siguiente tabla se muestra la puntuación de las diferentes métricas para cada uno de los algoritmos:

Resultados de las métricas						
	Modularidad	Coverage	Performance	RandIndex	TrueNeg	TruePos
Louvain	0.625	0.725	0.798	0.956	0.976	0.942
GirvanNewman k=3	0.389	0.987	0.571	0.681	0.222	1.000
GirvanNewman k=4	0.392	0.978	0.661	0.681	0.222	1.000
GirvanNewman k=5	0.392	0.977	0.676	0.657	0.246	0.942
GirvanNewman k=6	0.392	0.974	0.704	0.657	0.246	0.942
Label Propagation	0.389	0.987	0.571	0.681	0.222	1.000
Fluid Comm k=4	0.535	0.723	0.808	0.801	0.878	0.747
Fluid Comm k=5	0.553	0.582	0.831	0.692	0.976	0.495
Random k=4	-0.136	0.250	0.644	0.475	0.743	0.289

Tabla 4.2: Métricas de calidad. Manifestaciones Honduras, 2017

Los algoritmos que mejor puntúan en el índice de Rand son el de Louvain y Fluid Communities con *k=4*. Ambos proporcionan **resultados muy similares**. El resto de algoritmos no detectan una separación entre la comunidad roja y la naranja, y esto afecta en exceso al Ratio de Verdaderos Negativos.

4.4.3 Protestas en Nicaragua, 2018

El 18 de abril de 2018 comenzaron unas protestas en Nicaragua después de que el gobierno anunciase cambios en el sistema de pensiones. Las revueltas terminaron con al menos una treintena de muertos entre manifestantes y policías.

Se recolectaron alrededor de un millón de tweets enlazando artículos y noticias. Las **palabras clave** que se usaron para recolectar tweets fueron las siguientes: *Nicaragua, protestas, Daniel Ortega*

La siguiente figura muestra el grafo generado con las **400 noticias** más importantes en número de tweets:

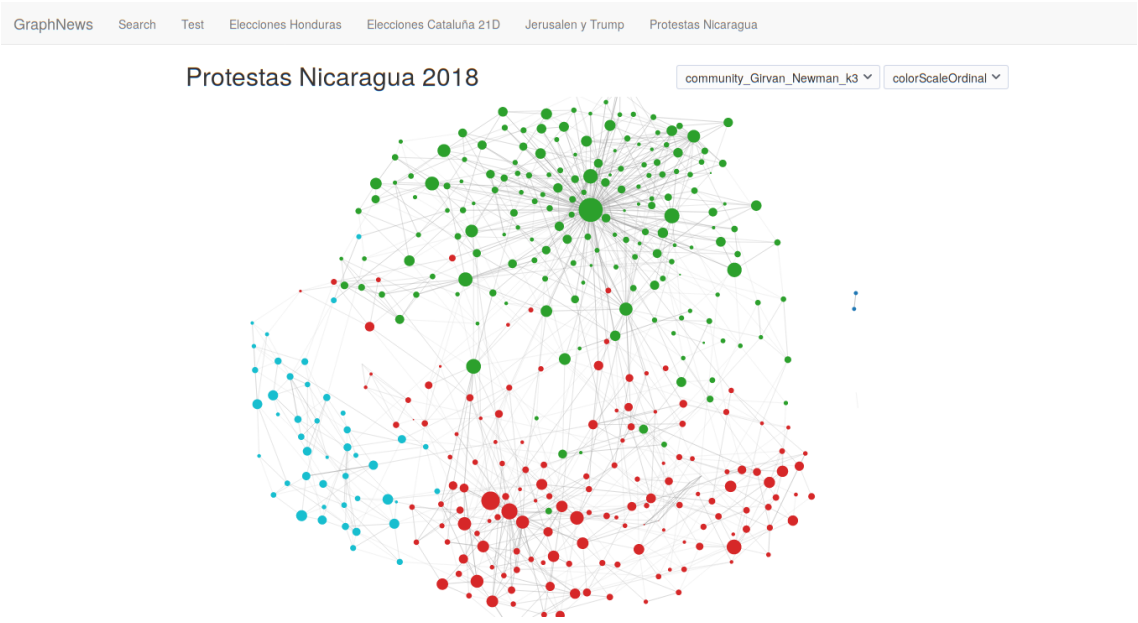

Figura 4.3: Protestas en Nicaragua. Se muestran las 400 noticias más twitteadas.

Comentarios tras las pruebas: El usuario manifestó las siguientes impresiones:

- El algoritmo Girvan Newman con *k=3* es el que mejor se comporta, si bien todos los algoritmos consiguen detectar las **3 comunidades** más importantes.
- La **comunidad verde** representa noticias a favor de los manifestantes, mientras que la **azul** representa noticias a favor del gobierno.
- La **comunidad roja** representa noticias de medios asociados a la Iglesia y al Vaticano. Esto tiene cierto sentido, ya que la iglesia ha actuado (y aún actúa) como mediadora en el conflicto. Esta comunidad también contiene la mayoría de noticias de medios internacionales en inglés.

En la siguiente tabla se muestra la puntuación de las diferentes métricas para cada uno de los algoritmos:

Resultados de las métricas						
	Modularidad	Coverage	Performance	RandIndex	TrueNeg	TruePos
Louvain	0.610	0.735	0.870	0.905	0.950	0.849
GirvanNewman k=3	0.485	0.922	0.621	0.971	0.962	0.982
GirvanNewman k=4	0.498	0.916	0.640	0.971	0.962	0.982
GirvanNewman k=5	0.500	0.911	0.659	0.971	0.962	0.982
Label Propagation	0.511	0.744	0.832	0.882	0.964	0.780
Fluid Comm k=3	0.482	0.870	0.664	0.890	0.877	0.907
Fluid Comm k=4	0.496	0.818	0.744	0.913	0.956	0.860
Random k=4	-0.054	0.229	0.738	0.527	0.757	0.240

Tabla 4.3: Métricas de calidad. Protestas Nicaragua, 2018

El usuario identifica el algoritmo de Girvan Newman con *k=3* como el que **mejores resultados** proporciona, coincidiendo con el índice de Rand. El resto de ejecuciones de Girvan Newman, con *k=4,5* producen resultados muy similares, únicamente añaden pequeñas comunidades que no afectan al valor del índice de Rand para los pares de noticias seleccionados.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

En esta sección se exponen las **conclusiones** extraídas tras el desarrollo de la herramienta y las pruebas realizadas. Se analizan las metas alcanzadas en este trabajo, con las cuales se logran los **objetivos propuestos**.

Las metas obtenidas son las siguientes:

- Se ha diseñado y desarrollado una herramienta que permite realizar una conexión con la API de Twitter, **descargar, filtrar y almacenar un gran número de tweets** de manera automatizada, al ritmo en que son producidos en tiempo real.
- La herramienta permite **construir un grafo de noticias** a partir de los tweets recopilados, en el que los enlaces se construyen a partir de las opiniones manifestadas por los usuarios en forma de "*likes*".
- La herramienta permite **analizar el grafo generado** de manera automática, incluyendo técnicas de simplificación, algoritmos de detección de comunidades y métricas de calidad.
- La herramienta proporciona al usuario una visualización del grafo a través de una **interfaz web** desarrollada para tal efecto. Esta interfaz cumple las condiciones de usabilidad requeridas.
- El sistema ha sido probado mediante **tres casos de uso reales**, y ha permitido analizar la distribución de grupos de opinión en las noticias, ayudando a identificar patrones de interés.

Las metas alcanzadas cumplen los objetivos impuestos al comienzo del proyecto. A continuación se exponen las posibles mejoras de la herramienta en base a los resultados obtenidos en la fase de pruebas.

Puntos positivos y a mejorar Los análisis realizados con noticias reales revelan que los grafos generados presentan estructuras de comunidades bastante evidentes, y que se puede extraer mediante los algoritmos utilizados. Sin embargo, se ha observado que el **tamaño del grafo** y el procesamiento previo que se realiza para simplificarlo influye en la

capacidad de detección de los algoritmos, aún cuando las comunidades detectadas sean en general las mismas. Este procesamiento previo influye también en la **calidad de la visualización** resultante, siendo la densidad de enlaces un factor limitante a la hora de poder identificar fácilmente las comunidades.

La herramienta construida ha logrado sus objetivos, ya que ha permitido analizar **múltiples casos distintos** sin necesidad de modificaciones. Además, el enfoque utilizado se ha demostrado capaz de identificar estructuras interesantes en el mapa de opinión de las noticias.

Un punto a mejorar es el **tiempo requerido** para recolectar toda la información que es necesaria para la creación del grafo, siendo el factor limitante la API pública de Twitter. Como trabajo a futuro se explorará un **enfoque predictivo** para anticipar la recolección de información, así como se analizarán las posibilidades que ofrece la API de pago.

5.1.1 Tecnologías aprendidas

Este trabajo me ha servido para aprender un gran número de tecnologías y herramientas, muchas de las cuales me eran desconocidas previamente.

En primer lugar, nunca había trabajado con esa cantidad de **datos**, y eso ha supuesto un aprendizaje en lo relativo a los detalles prácticos de manipulación de estos datos (manejo del servidor, base de datos, APIs), y al uso de técnicas de programación y algoritmos que puedan manejarlos de forma **eficiente**. Una parte de estos datos han sido procesados en tiempo real, lo que añade cierta complejidad.

En segundo lugar, la **teoría de grafos** era un área prácticamente nueva para mí, de la que conocía únicamente algunos conceptos básicos. Una parte importante del tiempo dedicado a este proyecto ha sido invertida en **investigación**.

El análisis de grafos, y su **aplicación en redes sociales** es un área puntera y de gran interés debido a las múltiples aplicaciones que tiene. En ese sentido este proyecto me ha servido para aprender las herramientas, librerías y tecnologías punteras más utilizadas en la actualidad, poniéndolas en práctica con un caso de uso real.

Por último, el proyecto me ha servido para poner en práctica y afianzar un gran número de **conocimientos** aprendidos a lo largo de la carrera, ya que **integra** componentes muy diversos. Entre ellos puedo destacar el manejo de bases de datos, conexión con APIs, mantenimiento del servidor, programación, diseño de software, tecnologías web backend y frontend, análisis de datos, teoría de grafos y estadística.

5.2 Trabajo futuro

Este trabajo puede servir de base para un proyecto más amplio.

Por un lado, podría **ampliarse** la cantidad de información disponible incorporando datos de **otras redes sociales** como Facebook, Tumblr o la propia Menéame.net. En estas plataformas existe un modelo de votación distinto al de Twitter, en las que el usuario puede evaluar una publicación tanto positivamente como negativamente (Facebook reactions permite incluso expresar algunos tipos de emociones).

Por otro lado, podría incorporarse el **análisis de texto**, tanto de las publicaciones que realizan los usuarios como de las propias noticias, incluir el **análisis de sentimientos**. Este tipo de análisis (especialemente aplicado a Twitter) está siendo bastante utilizado en los últimos años, con muy buenos resultados [30] [31].

En lo que respecta a la interfaz web, se van a explorar distintos **tipos de visualización**, que puedan hacer accesible la herramienta a usuarios sin conocimientos técnicos, y no habituados a trabajar con grafos.

Un objetivo básico para la continuación de este proyecto es la mejora de los **tiempos de respuesta**. Actualmente la API de Twitter supone el principal factor limitante ya que el número reducido de datos que proporcionan las búsquedas en la API REST obliga a usar la API Streaming. Esto reduce la utilidad de la herramienta, ya que obliga a esperar alrededor de 24 horas hasta que se recopilan suficientes datos y se puede empezar el análisis. Una solución a esto podría consistir en la utilización de la **API Enterprise** de Twitter. Por otro lado, y más interesante, sería la incorporación de algoritmos de predicción y **detección automática de sucesos** [25] [32] [26], en tiempo real, que permitirían comenzar la recolección de datos de manera anticipada.

BIBLIOGRAFÍA

[1] Rasmus Kleis Nielsen. Where do people get their news? https://medium.com/oxford-university/where-do-people-get-their-news-8e850a0dea03.

[2] Forbes México Rubén Vázquez. Diarios impresos vs. diarios digitales. https://www.forbes.com.mx/diarios-impresos-vs-diarios-digitales/.

[3] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.

[4] Ferran Parés, Dario Garcia-Gasulla, Armand Vilalta, Jonatan Moreno, Eduard Ayguadé, Jesus Labarta, Ulises Cortés, and Toyotaro Suzumura. Fluid communities: A competitive and highly scalable community detection algorithm.

[5] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3-5):75–174, 2010.

[6] Soroush Vosoughi, Deb Roy, and Sinan Aral. The spread of true and false news online. *Science*, 359(6380):1146–1151, 2018.

[7] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–36, 2017.

[8] Nic Newman. Mainstream media and the distribution of news in the age of social discovery. *Reuters Institute for the Study of Journalism, University of Oxford*, 2011.

[9] Alfred Hermida, Fred Fletcher, Darryl Korell, and Donna Logan. Share, like, recommend: Decoding the social media news consumer. *Journalism Studies*, 13(5-6):815–824, 2012.

[10] Eytan Bakshy, Solomon Messing, and Lada A Adamic. Exposure to ideologically diverse news and opinion on facebook. *Science*, 348(6239):1130–1132, 2015.

[11] Robert Huckfeldt, Paul Allen Beck, Russell J Dalton, and Jeffrey Levine. Political environments, cohesive social groups, and the communication of public opinion. *American Journal of Political Science*, pages 1025–1054, 1995.

[12] Elanor Colleoni, Alessandro Rozza, and Adam Arvidsson. Echo chamber or public sphere? predicting political orientation and measuring political homophily in twitter using big data. *Journal of Communication*, 64(2):317–332, 2014.

[13] Marco Pennacchiotti and Ana-Maria Popescu. A machine learning approach to twitter user classification. *Icwsm*, 11(1):281–288, 2011.

[14] Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7(Mar):551–585, 2006.

[15] Tien T Nguyen, Pik-Mai Hui, F Maxwell Harper, Loren Terveen, and Joseph A Konstan. Exploring the filter bubble: the effect of using recommender systems on content diversity. In *Proceedings of the 23rd international conference on World wide web*, pages 677–686. ACM, 2014.

[16] Eli Pariser. *The filter bubble: What the Internet is hiding from you*. Penguin UK, 2011.

[17] Lucas D Introna and Helen Nissenbaum. Shaping the web: Why the politics of search engines matters. *The information society*, 16(3):169–185, 2000.

[18] Seth Flaxman, Sharad Goel, and Justin M Rao. Filter bubbles, echo chambers, and online news consumption. *Public Opinion Quarterly*, 80(S1):298–320, 2016.

[19] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international conference on World Wide Web*, pages 271–280. ACM, 2007.

[20] BOE. Ley de protección intelectual. https://www.boe.es/boe/dias/2014/11/05/pdfs/BOE-A-2014-11404.pdf.

[21] Google Inc. Google news en españa. https://support.google.com/news/answer/6140047?hl=es.

[22] Meneame.net. Ley de protección intelectual. https://blog.meneame.net/tag/ley-propiedad-intelectual/.

[23] Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

[24] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

[25] Mario Cataldi, Luigi Di Caro, and Claudio Schifanella. Emerging topic detection on twitter based on temporal and social terms evaluation. In *Proceedings of the tenth international workshop on multimedia data mining*, page 4. ACM, 2010.

[26] Saša Petrović, Miles Osborne, and Victor Lavrenko. Streaming first story detection with application to twitter. In *Human language technologies: The 2010 annual conference of the north american chapter of the association for computational linguistics*, pages 181–189. Association for Computational Linguistics, 2010.

[27] Gangadharrao Soundaryarao Maddala and Kajal Lahiri. *Introduction to econometrics*, volume 2. Macmillan New York, 1992.

[28] Gert Sabidussi. The centrality index of a graph. *Psychometrika*, 31(4):581–603, 1966.

[29] Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

[30] Alexander Pak and Patrick Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, 2010.

[31] Efthymios Kouloumpis, Theresa Wilson, and Johanna D Moore. Twitter sentiment analysis: The good the bad and the omg! *Icwsm*, 11(538-541):164, 2011.

[32] Michael Mathioudakis and Nick Koudas. Twittermonitor: trend detection over the twitter stream. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pages 1155–1158. ACM, 2010.