UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA
SUPERIOR
Gº en Ingeniería Informática

TFG Ingeniería Informática:

Ububooknet

Presentado por Luis Miguel Cabrejas Arce
en Burgos el 1 julio de 2019
Tutores D. José Manuel Galán Ordax
y D. Luis Rodrigo Izquierdo Millán

D. José Manuel Galán Ordax y D. Luis Rodrigo Izquierdo Millán, profesores del departamento Ingeniería Civil, área de Organización de Empresas

Exponen:

Que el alumno D. Luis Miguel Cabrejas Arce, con DNI 71301637G, ha realizado el TFG Ingeniería Informática titulado: Ububooknet.

y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual, se autoriza su presentación y defensa.

En Burgos a 1 de julio de 2019

Vº. Bº del Tutor

Vº. Bº. del Tutor

D. José Manuel Galán Ordax

D. Luis Rodrigo Izquierdo Millán

Resumen

La motivación en el estudio de las redes complejas puede verse potenciado mediante el uso de conjuntos de datos adaptados a los intereses y motivaciones de las personas que se inician en el campo.

Ububooknet surge con el objetivo de permitir que cualquier persona, ya sea novel o experta en la materia, pueda generar una red de cómo interaccionan los personajes de cualquier novela en formato EPUB. Este software tiene aplicaciones en tres ámbitos diferentes: i) favoreciendo la docencia, consiguiendo que los alumnos se interesen más al aprender los conceptos inherentes al estudio de las redes complejas; ii) como herramienta de investigación en las Humanidades Digitales, permitiendo la comparación estructurada de las interacciones entre obras literarias, y iii) en sistema de recomendación, permitiendo extraer características interpretables de las novelas para su uso en sistemas basados en contenido

Descriptores

Generador de redes de interacción, visualización de la red, informes sobre la red, aplicación web, Python.

Abstract

The motivation in the study of complex networks might be boosted by the use of datasets adapted to the interest and motivations of the beginners in the field.

Ububooknet arises with the main goal of letting every kind of person, from novel to expert in the field, build his own character interaction network from his favourite book. This software has applications in three different areas: i) helping out the teaching, achieving a more interest in learning the inherent ideas of the complex networks study from the students side; ii) as an investigation tool in the Digital Humanities, allowing the structured comparison of the interactions between the literary works, and iii) in a recommending system, allowing the extraction of interpretable characteristics from the novels, for the use in content based systems.

Keywords

Interaction network generator, network displayer, reports about the network, web application, Python.

Índice General

Índice de figuras

A. Introducción

La ciencia de redes es un campo académico que estudia redes complejas que pueden ser de diversos tipos: de telecomunicaciones, informáticas, biológicas, semánticas y cognitivas, y sociales, considerando los actores representados por nodos y las conexiones o interacciones entre actores representadas como enlaces (1).

La ciencia de redes ha alcanzado una gran importancia en la sociedad actual; un ejemplo de su importancia es como la teoría de redes sociales fue crucial para la captura de Saddam Hussein (2). Pero detener personas no es la única utilidad que se puede alcanzar. Con la teoría de redes se pueden analizar grupos sociales para detectar cuáles pueden ser los principales focos de transmisión de enfermedades y de esta forma inmunizarles para evitar la propagación. Google utiliza la ciencia de redes a la hora de sugerir páginas cuando realizas una búsqueda. En el ámbito de la empresa se puede utilizar para encontrar la mejor opción a la hora de introducirte en un mercado, o para saber cómo difundir correctamente la información a tus empleados. En resumen, las aplicaciones que se puede dar a la ciencia de redes son muchas y muy importantes.

Actualmente, a la hora de aprender a extraer información sobre estas redes se suelen utilizar conjuntos de datos disponibles en Internet que pueden ser demasiado grandes para la correcta visualización, pueden tener formatos que no son leídos por la aplicación que se va a utilizar, pueden ser datos con un interés mínimo para el estudiante, o incluso pueden ser datos que no se sabe muy bien qué representan. Todo esto puede derivar en la pérdida de interés del estudiante por la materia.

Siendo este campo académico uno de gran importancia, es vital que el alumnado conserve el interés para así aprender más sobre la materia y poder aplicar estos conocimientos en el ámbito laboral o de la investigación. Este es uno de los motivos de los que parte este trabajo fin de grado.

Este proyecto permite a los usuarios generar y analizar desde un enfoque de red un conjunto de datos a partir de cualquier novela en formato EPUB. Desde una perspectiva docente, el software facilita adaptar el análisis y la obtención de datos a los intereses personales de lectura del alumno, que puede visualizarlos y obtener información sobre ellos desde la propia aplicación, y que se puede descargar para su uso en otras herramientas como pueden ser NetworkX, Pajek o Gephi.

Pero este trabajo no sólo es útil bajo el marco docente. El análisis sistemático de las características de las novelas tiene interesantes aplicaciones en el marco de las Humanidades Digitales (3). Por poner un ejemplo, nuestra herramienta permite el estudio sistemático, comparado y formal de las estructuras de interacción de personajes de los diferentes géneros, estilos y épocas de la literatura universal. Además, esta información resulta relevante para su uso en sistemas de recomendación. El software puede entenderse como una herramienta de *feature extraction* de las novelas, en las que las propiedades obtenidas de la red permiten personalizar y adaptar las recomendaciones a los usuarios. Si bien los filtros colaborativos han tenido mucho

éxito en el proceso de recomendación de novelas, películas, productos, etc (4), los filtros basados en contenidos, en los que las características de análisis son interpretables aportan ventajas, no sólo en la recomendación, sino potencialmente en el propio diseño y adaptación de las novelas.

1.1. Estructura de la memoria

La memoria sigue la siguiente estructura:

- **Introducción:** Breve descripción del contenido del trabajo. Estructura de la memoria y lista de materiales adjuntos.
- **Objetivos del proyecto:** Exposición de los objetivos que persigue el proyecto.
- **Conceptos teóricos:** Exposición de los conceptos clave para el desarrollo y comprensión del proyecto.
- **Técnicas y herramientas:** Listado de metodologías y herramientas utilizadas para la correcta resolución del proyecto.
- **Aspectos relevantes del desarrollo:** Exposición de aspectos relevantes surgidos durante la realización del proyecto.
- **Trabajos relacionados:** Breve resumen de estudios ya realizados relacionados con el proyecto.
- **Conclusiones y líneas de trabajo futuras:** Conclusiones obtenidas tras el desarrollo del proyecto. Análisis crítico del proyecto sugiriendo posibilidades de mejora y de expansión de la solución.

Junto a la memoria se proporcionan los siguientes anexos:

- **Planificación del proyecto:** Planificación temporal y estudio económico.
- **Especificación de requisitos:** Objetivos generales, catálogo de requisitos y especificación de requisitos.
- **Especificación de diseño:** Se describen el diseño de los datos, el diseño de paquetes y el diseño de interfaz.
- **Manual del programador:** Se describe la estructura de directorios, instalación de herramientas, ejecución de la aplicación y de las pruebas.
- **Manual de usuario:** Guía para el correcto uso de la aplicación por parte del usuario.

1.2. Materiales adjuntos

Los materiales adjuntos junto con la memoria son:

- **Repositorio del proyecto:**
  - https://github.com/lca0037/GII18.0U-Ububooknet

- **Despliegue de la aplicación:**
  - http://ububooknet.herokuapp.com/

B. Objetivos del proyecto

En esta sección se detallan los diversos objetivos que se busca conseguir a la hora de realizar el proyecto.

2.1. Objetivos generales

- Desarrollar una aplicación web que permita a los usuarios generar una red de interacciones entre los personajes de un documento en formato EPUB.
- Desarrollar herramientas que permitan al usuario personalizar los datos a obtener.
- Facilitar la visualización de la red obtenida y la personalización de esta sin costes elevados de tiempo.
- Ofrecer mayor información sobre los datos obtenidos mediante informes personalizables.

2.2. Objetivos técnicos

- Utilizar analizadores léxicos para la creación automática de un diccionario de personajes y para encontrar las posiciones de los personajes del diccionario en el texto.
- Integrar el proyecto network_styling_with_d3 para la visualización de la red generada.
- Permitir al usuario arrastrar personajes para juntarlos.
- Permitir el acceso concurrente a la aplicación.
- Internacionalizar la aplicación de forma que pueda ser accedida y entendida por personas que hablen inglés y/o español.
- Utilizar Python para generar las redes de interacciones.
- Utilizar Flask para el desarrollo web.
- Utilizar NetworkX para el análisis de los datos generados.
- Utilizar el patrón de diseño Modelo-Vista-Presentador
- Utilizar GitHub como herramienta de control de versiones.
- Proteger la información sensible almacenada en la aplicación frente a posibles ataques.
- Realizar tests que garanticen la calidad del código.
- Aplicar la metodología ágil Scrum durante el desarrollo del software.
- Utilizar ZenHub como herramienta para la gestión de proyectos.
- Hacer disponible la aplicación a través de la web.

2.3. Objetivos personales

- Aumentar conocimientos sobre el desarrollo web.
- Facilitar el trabajo docente.
- Aplicar conocimientos adquiridos durante el grado.
- Adquirir la experiencia de desarrollar un proyecto software desde cero utilizando metodologías ágiles.

C. Conceptos teóricos

3.1. EPUB

EPUB es un formato de libro electrónico que se convirtió en el formato estándar del Foro Internacional de Publicaciones Digitales (IDPF) en el año 2007 y sigue vigente hasta la fecha. (5)

La última gran versión es EPUB 3; esta versión consiste en cuatro especificaciones que definen diferentes aspectos importantes sobre las publicaciones EPUB. Estas son:

1. EPUB Publications, que define las semánticas a nivel de publicación y los requerimientos de conformidad general para publicaciones EPUB.
2. EPUB Content Documents, que define los perfiles de XHTML, SVG y CSS para su uso en las publicaciones EPUB.
3. EPUB Open Container Format, que define el formato del fichero y un modelo de procesado para el encapsulamiento de una serie de recursos relacionados en un solo fichero.
4. EPUB Media Overlays, que define un formato y un modelo de procesamiento para la sincronización de texto y audio.

Los recursos de las publicaciones están normalmente empaquetados como un archivo basado en ZIP, que tiene una extensión ".epub". El Container Format provee una forma de determinar que el contenido de un fichero ZIP representa a una publicación EPUB, y también provee un directorio con un nombre predefinido que posee recursos informativos ("/META-INF").

El fichero clave de este directorio es "container.xml" que dirige al sistema de lectura al archivo raíz de la publicación, el Package Document. El Package Document, conocido como "EPUB Navigation Document", especifica todos los documentos de contenido que constituyen la publicación y los recursos que necesitan, además de un orden de lectura por defecto; este es un documento XHTML con una extensión ".opf" (también puede ser un archivo ".ncx" que se utiliza por compatibilidad con versiones anteriores de EPUB). Este fichero contiene una etiqueta conocida como "spine" que contiene el orden a seguir. Los EPUB también contienen uno o varios ficheros conocidos como EPUB Content Documents, que son ficheros de formato XHTML o SVG que describen el contenido leíble de una publicación y hacen referencia a los archivos multimedia asociados a estos. (6)

3.2. Redes complejas

Para entender lo que es una red compleja lo primero que debemos de saber es qué es una red y algunos conceptos básicos sobre estas. Una red o grafo representa un conjunto de nodos y sus enlaces (relaciones). En una red los enlaces pueden representar

cualquier tipo de información, pero todos los enlaces deben representar lo mismo en toda la red, y en función de lo que representen solo se podrán responder ciertos tipos de preguntas. Los nodos pueden ser de una sola clase, que serían las redes unimodales, de dos clases, que serían redes bimodales, o de múltiples clases, que serían las redes multimodales. En nuestro caso la red que generemos será una red unimodal porque cada nodo se corresponde con un personaje. Los enlaces también pueden ser de distintos tipos. Pueden ser dirigidos o no dirigidos; los dirigidos tienen una relación origen destino mientras que los no dirigidos tienen una relación simétrica. Los enlaces dirigidos con el mismo nodo origen y destino son conocidos como auto-enlaces. Las redes a su vez también pueden ser de distintos tipos: si es simple, solo hay un enlace como máximo por cada par de nodos; en caso contrario es múltiple. Si los enlaces de la red representan únicamente la existencia del tipo de relación serán binarias, y en caso de que los enlaces tengan asociado además un peso serán pesadas. En nuestro caso, la red generada tiene enlaces no dirigidos, y es una red simple y pesada. Una vez conocidos estos conceptos básicos, podemos decir que una red compleja es aquella compuesta por una gran cantidad de nodos y cuyo patrón de conexiones no es regular. (7,8)

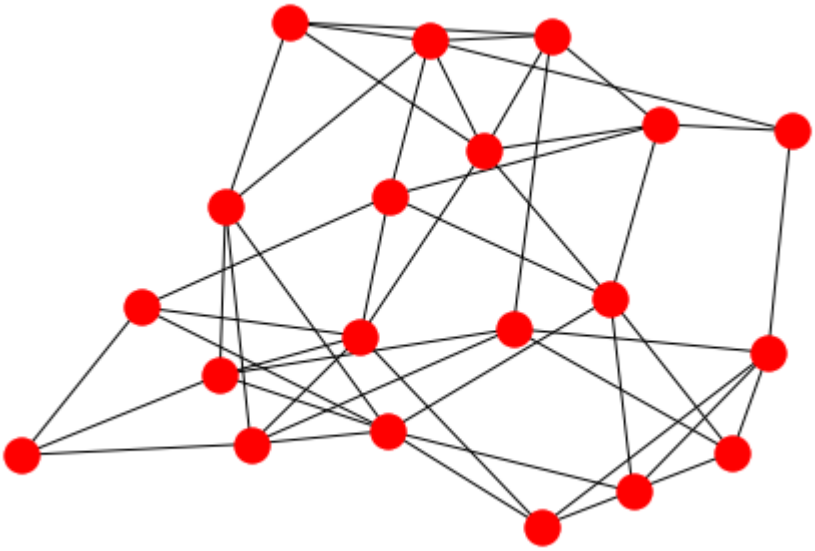

Figura C 1 Ejemplo de red compleja

3.3. Grado de los nodos

El grado de un nodo (k) en una red no dirigida es el número de enlaces que tiene conectados, si no tenemos en cuenta el peso de los enlaces, o el sumatorio del peso de los enlaces conectados a un nodo en caso de tenerlo en cuenta. Los nodos con un alto grado en comparación a la media se conocen como hubs. (7,8)

Se puede calcular el grado medio de una red no dirigida de la siguiente forma:

$$\langle k \rangle = \frac{2L}{N}$$

Donde:

N es el número de nodos

L es el número de enlaces

Distribución de grado

Con el grado de todos los nodos podemos obtener la distribución de grado P(k), que es la fracción de nodos que tienen grado k. (7,8)

Esta fórmula se representa como:

$$P(k) = {}^{N_K}/_{N}$$

Donde N_k es igual al número de nodos con grado k.

Densidad

La densidad de la red es la relación entre el número de enlaces y el número posible de enlaces. Se dice que una red es poco densa en caso de que el número de enlaces L es del mismo orden que el número de nodos N. Si por el contrario L >> N se considera que la red es densa. (7,8)

3.4. Distancia geodésica

En las redes, la distancia geodésica entre dos nodos es el mínimo número de enlaces que hay que atravesar para llegar desde el nodo origen al nodo fin. En redes pesadas, la distancia sería la mínima suma de los pesos de los enlaces que hay que atravesar. En caso de no existir ningún camino entre dos nodos se considera que la distancia geodésica es infinita. (7,8)

Excentricidad

Se le llama excentricidad a la mayor distancia geodésica que tiene un nodo con respecto al resto de nodos de la red. (7,8)

Diámetro de la red

El diámetro es la excentricidad máxima entre todos los nodos de la red. Los nodos con excentricidad máxima se denominan nodos periféricos. El diámetro nos proporciona el número de pasos máximos necesarios para ir de cualquier nodo a cualquier otro de la red y es una de las principales medidas de conectividad global de una red. (7,8)

Radio de la red

El radio es la excentricidad mínima entre todos los nodos de la red. Los nodos con excentricidad mínima forman el centro. Dos veces el radio de la red siempre es mayor o igual que el diámetro de la red. (7,8)

3.5. Coeficiente de clustering

El coeficiente de clustering mide la densidad local de la red; esto lo hace calculando la proporción de vecinos de cada nodo que están conectados entre sí. Un bajo coeficiente de clustering de un nodo puede indicar que sus vecinos dependen más de él para obtener información; esto hace que pueda usarse como una medida de centralidad inversa en sustitución de la medida de intermediación (la veremos posteriormente en las medidas de centralidad) (7,8).

$$C_i = \frac{2L_i}{k_i(k_i - 1)}$$

Una medida global de la red frecuentemente utilizada para caracterizarla es precisamente el coeficiente de clustering local medio.

$$\langle C \rangle = \frac{1}{N}\sum_{i=1}^{N} C_i$$

Transitividad

La transitividad mide el número de triadas abiertas que son triángulos en la red. Para calcularla se utiliza la siguiente fórmula:

$$Transitividad = \frac{3 \times Nº\ de\ tríadas\ cerradas}{Nº\ de\ tríadas\ abiertas}$$

A pesar de que tanto la transitividad como el coeficiente de clustering local medio miden la tendencia de los enlaces a formar triángulos, la transitividad da una mayor importancia a los nodos que tengan un grado mayor (7,8).

3.6. Medidas de centralidad

Las medidas de centralidad sirven para encontrar los nodos más importantes de una red. Hay distintas formas de medir la importancia; se puede calcular por cada nodo, lo

que supondría la centralidad de nodo, o, si la centralidad se calcula para la red en su conjunto, se conoce como medida de centralización y mide la desigualdad o dispersión que tiene una determinada medida de centralidad en la red. Existen también diferentes métricas de centralidad (7,8).

Centralidad de grado

La centralidad de grado es una medida local que parte de la hipótesis de que los nodos con más grado son los más importantes de la red. El fallo de esta métrica es que depende exclusivamente del grado sin tener en cuenta la importancia de los nodos con los que tienes un enlace. De acuerdo con esta métrica, el hecho de que en una red social te siga un bot sería igual de importante a que te siga alguien famoso (7,8).

Centralidad de cercanía

La centralidad de cercanía parte de la hipótesis de que los nodos que están más cerca de otros nodos son los más importantes. Para determinar cuáles son los nodos más cercanos se utiliza la *closeness* de cada nodo, que consiste en el inverso de la suma de las distancias geodésicas al resto de nodos, normalizadas. Los nodos más importantes son los nodos con mayor *closeness.*

Algunos problemas que surgen con esta medida son que no discrimina mucho en redes pequeño mundo (redes donde se puede llegar de cualquier con una distancia relativamente corta) al haber poca diferencia entre la distancia geodésica media menor y mayor, y que en caso de que haya distancia infinita entre varios nodos, la *closeness* de estos también será infinita. Una solución al caso de las distancias infinitas es utilizar la distancia geodésica media armónica, cuya fórmula es: (7,8).

$$C'_i = \frac{1}{N-1}\sum_{j\neq i}^{N}\frac{1}{d_{ij}}$$

Centralidad de intermediación

La centralidad de intermediación asume que los nodos que conectan nodos son los nodos importantes. Para calcular la intermediación de un nodo se utiliza la siguiente fórmula: (7,8)

$$x_i^{bwn} = \sum_{st(s\neq t)}\frac{n_{st,i}}{g_{st}}$$

Donde g_{st} es el número de caminos más cortos entre s y t, cuando no haya caminos la división dará como resultado 0 y $n_{st,i}$ que será 1 si i es un camino corto entre s y t, y 0 en caso contrario.

El problema de esta medida es el alto coste computacional que supone calcularla.

Centralidad de valor propio

Esta medida de centralidad se basa en la hipótesis de que la importancia de un nodo crece si tiene vínculos que también son importantes. Para esto utiliza un algoritmo que inicializa a todos los nodos con una importancia de valor 1, y propaga recursivamente la importancia a través de los enlaces (7,8).

El vector de centralidad de los nodos X satisface que:

$$AX = k_1 X$$

$$x_i = k_1^{-1} \sum_{j=1}^{n} A_{ij} \times x_j$$

Siendo A la matriz de adyacencia.

La centralidad del nodo i se corresponde con el elemento x_i^1 del autovector principal (mayor autovalor).

Dadas t iteraciones:

$$C_e(t) = \lambda_1^t \sum_i \alpha_i \left(\frac{\lambda_i}{\lambda_1}\right)^t v_i$$

Siendo λ_1 el mayor autovalor.

Pagerank

Pagerank es un algoritmo que parte de la centralidad de valor propio. Las modificaciones que incluye son que la importancia que un nodo recibe de sus vecinos es proporcional a su centralidad divida por el número de nodos a los que aporta importancia, y añade dos parámetros para calcular la importancia: uno es beta que es una constante positiva para garantizar el valor positivo no nulo de todos los nodos de la red, y el otro es alfa que modula la importancia con respecto a la centralidad de un nodo (7,8).

$$x_i = \alpha \sum_{j=1}^{N} A_{ij} \frac{x_j}{d_j^{out}} + \beta$$

3.7. Grupos y comunidades

La mayoría de las redes se dividen naturalmente en grupos o comunidades, como pueden ser grupos de amigos, compañeros de trabajo, familia, etc. y estos grupos y comunidades se manifiestan en los enlaces entre los nodos. Hay diferentes métricas y algoritmos para encontrar estos grupos o comunidades que van a ser explicadas a continuación (7,8).

Cliques

Un clique es un subconjunto de nodos que está completamente conectado entre ellos, es decir, cada nodo tiene al menos un enlace con los demás nodos del clique. El mayor problema de buscar cliques es la alta carga computacional requerida al tratarse de un problema NP-completo (7,8).

K-plex

El requisito de los cliques es muy exigente y es poco próximo a las redes sociales reales; esto hace que surja la solución de k-plex que propone que un subconjunto de nodos de tamaño n se considere k-plex si cada nodo está conectado a al menos n-k nodos del subconjunto (7,8).

K-core

Un k-core es un subconjunto máximo de nodos en el que cada uno de ellos está conectado al menos a k nodos del subconjunto. Este método no permite que dos nodos pertenezcan a más de un k-core a la vez. Su algoritmo consiste en ir eliminado los nodos con un grado menor que k hasta encontrar el k-core de mayor k (7,8).

K-clique

Un k-clique es un clique en el que el subconjunto de nodos no tiene que estar directamente relacionado con el resto de nodos del subconjunto sino que se permite que estén a una distancia menor o igual que k. Una alternativa a k-clique es k-clan en el que además los caminos que unen a los nodos deben incluir solo nodos que pertenecen al mismo subconjunto (7,8).

K-componente

Un k-componente es un subconjunto de nodos para los que todos los nodos están conectados al resto de nodos del subconjunto por k caminos independientes como mínimo (7,8).

Algoritmos de detección de comunidades

Se definen las comunidades como grupos de nodos con una densidad de conexión alta entre ellos y que a su vez tienen pocas conexiones con otros grupos. Para particionar una red en comunidades existen distintos métodos, los aglomerativos, los

de eliminación de enlaces, los que maximizan la modularidad y otros tipos de métodos (9,10).

Modularidad

La modularidad de una partición es el sumatorio de la fracción de los enlaces que hay dentro de cada comunidad entre el número de enlaces total de la red, menos la fracción al cuadrado del número de enlaces que van a otras comunidades entre el número de enlaces total de la red, que en notación matemática se entiende como (11):

$$Q = \sum_{c} [e_{ii} - a_i^2]$$

Donde:

$$e_{ii} = \sum_{ij} \frac{A_{ij}}{2m} \delta(c_i, c_j)$$

$$a_i = \sum_{j} e_{ij}$$

Siendo e_{ii} la fracción de enlaces con ambos vértices en la misma comunidad y a_i la fracción de enlaces unidos a vértices de la comunidad i.

3.8. Detección de roles

La detección de roles parte de la idea de que nodos con un mismo rol deben de tener propiedades similares en la red (12). Antes de detectar el rol de cada nodo lo primero que debe hacerse es determinar comunidades en la red; una vez determinadas, se analiza cómo está posicionado el nodo en su propia comunidad y su relación con respecto al resto de comunidades.

Para determinar el rol de cada nodo primero debe hallarse el grado dentro de la comunidad y el coeficiente de participación del nodo.

El grado respecto a la comunidad se calcula como:

$$z_i = \frac{k_i - \bar{k}_{s_i}}{\sigma_{k_{s_i}}}$$

Siendo $\bar{k}_{si}$ la media del grado de los nodos de la comunidad a la que pertenece el nodo i, k_i el grado del nodo i respecto a la comunidad a la que pertenece el nodo, y σ_{ksi} la desviación de k_i con respecto a $\bar{k}_{si}$.

Y el coeficiente de participación se calcula como:

$$P_i = 1 - \sum_{s=1}^{N_M} \left(\frac{k_{is}}{k_i}\right)^2$$

Siendo k_i el grado del nodo i respecto a la red completa y k_{is} el grado del nodo i con respecto a una comunidad s.

Para determinar el rol lo primero que se hace es dividir los nodos en hubs si su z es mayor o igual a 2.5 y como no hub en caso contrario (7,8,12)

.

Los roles de nodos no hubs son los siguientes:

1. Nodos ultra-periféricos: tienen todos los enlaces en su comunidad; su P es aproximadamente 0.
2. Nodos periféricos: tienen al menos el 60% de sus enlaces en su comunidad; su P es menor que 0.625
3. Nodos no hub conectores: tienen aproximadamente la mitad de los enlaces dentro de la comunidad; su P es mayor o igual que 0.625 y menor que 0.8.
4. Nodos no hub kinless: Son nodos que tienen menos del 35% de sus enlaces dentro de la comunidad. Estos nodos son nodos que no pueden ser asignados de forma clara a ninguna comunidad; su P es mayor o igual que 0.8.

Los roles de nodos hub son los siguientes:

1. Nodos provinciales: Son nodos que tienen al menos 5/6 de sus enlaces dentro de la comunidad; su P es menor o igual que 0.3.
2. Nodos hubs conectores: Son nodos que tienen aproximadamente la mitad de sus enlaces dentro de la comunidad; su P es menor que 0.75 y mayor o igual que 0.3.
3. Nodos hubs kinless: Al igual que con los nodos que no son hubs, no se les puede asignar de forma clara a ninguna comunidad; su P es mayor o igual que 0.75.

3.9. Análisis léxico

El análisis léxico es el proceso de convertir una secuencia de caracteres en unos componentes léxicos (tokens). Los programas que realizan este proceso son conocidos como lexers, y suelen ser combinados con un parser (como en este caso no lo utilizamos, no va a ser explicado) (13).

Los analizadores léxicos están compuestos por patrones, que son la forma que tienen que tomar los lexemas para ajustarse a los tokens, los patrones estarán formados por expresiones regulares; lexemas, que son una secuencia de caracteres comprendidos únicamente por un token; y tokens, que son la unidad mínima de información con algún tipo de significado. Cuando el analizador léxico encuentra una coincidencia con el patrón de un token, este devuelve una serie de atributos como pueden ser el lexema y el tipo de token encontrado, y con estos datos podemos realizar una serie de acciones (14).

Sin ahondar mucho en su funcionamiento, el analizador léxico posee una máquina de estados finita, que nos puede servir para cambiar de un estado a otro, que puede

tener nuevos tokens, o utilizar los mismos, u otros con el mismo patrón pero con acciones diferentes (13,15).

3.10. Web scraping

El *web scraping* es una técnica utilizada mediante programas software, que simulan la navegación de un humano, para extraer información de sitios web (16).

El motivo principal para realizar *web scraping* es obtener grandes cantidades de información de una página web de manera completamente automática. Para poder obtener estos datos se puede hacer mediante diversas herramientas que hay en la web, aunque estas pueden no ser lo suficientemente personalizables para realizar el trabajo, como puede ser utilizando webscraper.io, o se puede analizar el HTML de las páginas web y extraer la información directamente de este HTML mediante alguna herramienta que facilita la extracción de datos de código HTML como puede ser BeautifulSoup, o directamente mediante expresiones regulares (17).

D. Técnicas y herramientas

En esta sección se explican las diversas técnicas y herramientas empleadas para realizar la aplicación.

4.1. Metodología ágil - Scrum

- **Motivación**: Metodología aprendida durante el grado.

Scrum es el nombre con el que se denomina a uno de los marcos de desarrollo ágiles. También es un proceso en el que se aplican un conjunto de buenas prácticas de manera regular, para trabajar colaborativamente y de esta forma obtener el mejor resultado posible a la hora de desarrollar un proyecto (18).

Para alcanzar el objetivo, se utiliza el desarrollo iterativo e incremental en sustitución de la planificación y ejecución completa del producto. La calidad del resultado se basa en el conocimiento tácito de las personas en equipos auto-organizados, más que en la calidad de los procesos empleados, y en el solapamiento de distintas fases del desarrollo que en la utilización de un ciclo secuencial o en cascada (18).

4.2. Herramienta de control de versiones

- **Elección:** Git - GitHub
- **Motivación:** Herramienta utilizada previamente en otras asignaturas del grado.

Git es un software de control de versiones diseñado para manejar todo tipo de proyectos, desde grandes a pequeños proyectos, con eficiencia y velocidad (19).

GitHub es una plataforma donde se permite guardar código, que utiliza Git para el control de versiones y la colaboración entre miembros del proyecto (20).

4.3. Herramienta de gestión de proyectos

- **Elección:** ZenHub
- **Motivación:** Herramienta utilizada previamente en otras asignaturas.

ZenHub es una extensión de navegador que añade utilidades de manejo de proyectos directamente en la interfaz de usuario de GitHub, haciendo que la colaboración sea más rápida y visual, y más ordenada. Se puede utilizar para planificar sprints, crear tareas épicas, y visualizar el “workflow” del proyecto sin tener que salir de GitHub (21).

4.4. Gestor de referencias bibliográficas

- **Elección:** Zotero
- **Motivación:** Se utiliza Zotero por su integración con el editor de texto utilizado para la realización de la memoria, y por la familiaridad adquirida mediante el uso de este gestor de referencias en otras asignaturas (22).

Zotero es un software para la gestión de referencias, de una gran comodidad de uso al poder añadir automáticamente referencias a contenido disponible en la web desde la extensión que hay para navegadores como Google Chrome o Mozilla Firefox, y que permite la exportación de la bibliografía generada a múltiples formatos de citas (23).

- **Alternativa:** Mendeley

Mendeley es un software de unas características similares, que también se puede integrar con el editor de texto utilizado y con una facilidad de agregación de referencias desde el navegador también existente (24,25).

4.5. Herramienta de prototipado de interfaces

- **Elección:** Pencil Proyect
- **Motivación:** Familiaridad de uso al haberla utilizado en otras asignaturas del grado.

Pencil es una herramienta de prototipado de interfaces open-source y gratuita, que permite crear interfaces web, de escritorio y móvil (26,27).

- **Alternativa:** AdobeXD

AdobeXD es una herramienta que también se puede usar de forma gratuita, aunque solo se puede utilizar para prototipado de aplicaciones móviles o de páginas web. No es tan fácil de usar como Pencil (28).

4.6. Lenguaje de programación

- **Elección:** Python
- **Motivación:** Gran popularidad del lenguaje de programación y experiencia previa

4.7. Librerías y módulos de Python

En esta sección se nombran y explican algunas librerías y módulos propios de Python utilizados en la aplicación y que pueden ser interesantes.

Zipfile

- **Motivación:** Los EPUBs son archivos zip, por lo que se necesita software para descomprimir los archivos y acceder al contenido.

Zipfile es una librería que permite trabajar con archivos zip, ya sea para leerlos como para crearlos o escribirlos; también permite comprobar si el archivo indicado es realmente un archivo zip o no (29).

Shutil

- **Motivación:** Se necesita borrar directorios que pueden contener archivos en su interior.

Shutil es un módulo que permite realizar operaciones de alto nivel sobre archivos y grupos de archivos (30).

Secrets

- **Motivación:** Se necesitan generar claves aleatorias para encriptar las cookies y para generar números en hexadecimal para generar colores aleatorios.

Secrets es un módulo que se usa para generar contraseñas criptográficamente fuertes para proteger datos sensibles (31).

4.8. Cobertura del código

- **Elección:** Unittest
- **Motivación:** Facilidad de uso a la hora de generar test unitarios.

4.9. Generación de la red

- **Elección:** NetworkX
- **Motivación:** Paquete por excelencia a la hora de generar redes complejas en Python.

NetworkX es un paquete de Python para la creación, manipulación, y estudio de las estructuras dinámicas y funciones de las redes complejas. NetworkX proporciona una gran cantidad de métodos y algoritmos para representar redes (32).

4.10. Representación de la red

- **Elección:** network styling with d3
- **Motivación:** Permite al usuario interactuar con la red pudiendo visualizarla de muy diversas formas.

Esta aplicación web permite recibir los parámetros de la red por url y también permite al usuario subir directamente los datos de la red (en nuestra aplicación esto se modifica para que reciba directamente la red sin necesidad de pasarle los datos por url o que el propio usuario tenga que hacerlo). Posteriormente ofrece al usuario una serie

de opciones de personalización de la red. Esta aplicación web es la misma que utiliza netwulf para representar las redes generadas en NetworkX (33,34).

- **Alternativa:** Representación propia de NetworkX

Aunque la representación propia de NetworkX también nos permite utilizar distintos tamaños de nodos, e incluso representar nodos con distintos colores, estas opciones se consiguen modificando el código Python que genera la imagen de la red. A pesar de no utilizar esta representación de forma principal, sí que va a ser utilizada cuando el usuario solicite informes sobre la red.

4.11. Lectura de EPUBs

- **Elección:** BeautifulSoup4 + zipfile
- **Motivación:** Gran documentación sobre las librerías.

BeautifulSoup4 es una librería que provee de una serie de herramientas que facilitan la obtención de datos de archivos o textos con el formato HTML, XML y XHTML, que son los formatos de los archivos que componen el EPUB. Zipfile, como hemos comentado previamente, nos permite leer los archivos del EPUB, al ser el EPUB al fin y al cabo, un archivo zip (35).

- **Alternativa:** Ebooklib

Se descarta debido a la pobre documentación de la librería.

4.12. Scraping

- **Elección:** BeautifulSoup4 + urllib
- **Motivación:** Experiencia previa con la librería BeautifulSoup4.

Urllib es un módulo de Python para trabajar con URLs. Este módulo está compuesto por cuatro paquetes, que son (36):

- urllib.request: sirve para abrir y leer URLs
- urllib.error: contiene las excepciones que produce urllib.request
- urllib.parse: sirve para parsear URLs
- urllib.robotparser: sirve para parsear achivos robots.txt

4.13. Analizador léxico

- **Elección:** Ply
- **Motivación:** Similitud con herramientas utilizadas en asignaturas del grado.

Ply es una librería que implementa de manera fiel las herramientas de lex y yacc en Python, lo que permite construir un lexer o un parser de forma relativamente sencilla si se conocen previamente las herramientas anteriores (37).

4.14. Interfaz gráfica

- **Elección:** Flask
- **Motivación:** Interés en aprender a desarrollar aplicaciones web.

Flask es un micro framework de Python para el desarrollo web, basado en el Web Service Gateway Interface (WSGI) de Werkzeug y el motor de plantillas de Jinja2 (38).

Además de ser sencillo de usar gracias a diversos tutoriales que pueden ser encontrados en la web (39), Flask dispone de diversas extensiones que facilitan la vida del programador. Una de ellas, que también es usada en la aplicación, es Flask babel que nos permite internacionalizar fácilmente nuestra página web (40).

4.15. Programación en el lado del cliente

- **Elección:** JavaScript + jQuery
- **Motivación:** JQuery simplifica labores no triviales en la programación web.

JavaScript es un lenguaje de programación que todos los navegadores modernos pueden interpretar y que nos permite la modificación de páginas web gracias a la capacidad de interaccionar con el DOM que se provee al lenguaje (41).

JQuery es una biblioteca multiplataforma de JavaScript que simplifica tanto la manipulación del árbol DOM, como el manejo de eventos, y las llamadas AJAX, que facilitan la comunicación entre el cliente y el servidor (42).

E. Aspectos relevantes del desarrollo del proyecto

En esta sección se van a detallar las partes más importantes en el desarrollo del proyecto, desde la toma de decisiones y sus implicaciones, a los problemas afrontados y las soluciones a estos.

5.1. Metodologías

Para el desarrollo del proyecto se siguió una metodología ágil, en concreto Scrum, pero con algunas diferencias al desarrollarse el proyecto en un entorno educativo y no de empresa. Los puntos que sí que se siguieron a la hora de realizar el proyecto fueron:

- Se realiza un desarrollo incremental; al final de cada sprint se disponía de una parte del proyecto operativa.
- Se realiza una reunión al final de cada sprint donde se revisa el incremento de funcionalidades generado y se planifica el nuevo sprint.
- Al final de cada reunión se añaden nuevas funcionalidades a realizar a la pila del sprint.
- A las funcionalidades se les estima un tiempo de realización.
- La duración media de los sprints es de 2 semanas.
- Cuando una funcionalidad del sprint se implementa se actualiza la pila del sprint para monitorizar el avance del sprint.

5.2. Formación

Para la realización del proyecto se requería una serie de conocimientos de los que no disponía desde un principio. Esta falta de conocimientos era mayoritariamente sobre desarrollo web, y sobre redes complejas.

Para adquirir los conocimientos necesarios para trabajar con Flask se siguieron los siguientes tutoriales:

- Flask Tutorial Web Development with Python (43)
- The Flask Mega Tutorial (39)

Para la formación en redes complejas, sin embargo, no se siguió ningún tutorial debido a que iba a ser cursada durante el desarrollo del proyecto.

Cabe mencionar que tampoco se conocía cómo estaba estructurado por dentro un archivo de tipo EPUB. Esto no supuso un gran problema debido a la relativa simpleza

de su estructura y a la facilidad de uso de la librería utilizada para extraer la información necesaria.

Las conclusiones que se pueden extraer de la formación es que a la hora de hacer un proyecto es casi imposible conocer todas las librerías y herramientas necesarias para todas las funcionalidades. Es importante elegir una librería con una buena documentación, debido a que seguramente se va a tener que recurrir a ella en algún punto del desarrollo, y que además sea conocida; cuanta más gente use la librería, más probable es que las dudas que te puedan surgir le hayan surgido previamente a otra persona y ya hayan sido resueltas.

Es por estos motivos por los que dedicar un tiempo a buscar librerías de calidad, puede ahorrar después mucho tiempo.

5.3. Establecimiento de requisitos

En la primera reunión del proyecto se definen una serie de funcionalidades que se han de implementar a lo largo del desarrollo del proyecto. Se parte de una serie de requisitos que en principio eran sencillos y que se comprendían correctamente que son los siguientes:

- Leer un EPUB y almacenar la información relevante.
- Generar un diccionario buscando las palabras en mayúsculas, importando un csv o haciendo scraping.
- Modificar el diccionario por el usuario
- Buscar la posición de los nombres que forman el diccionario
- Crear de la red
- Analizar, visualizar, generar informes y exportar de la red.

El problema parte de la falta de especificación en la generación de los requisitos debida a la costumbre de tener unos requisitos muy claros a la hora de hacer prácticas en las asignaturas. Este problema derivó en altos costes de tiempo a la hora de cambiar el código.

Un ejemplo muy claro de los problemas derivados fue el hecho de tener en cuenta si se está en un mismo capítulo o no a la hora de interaccionar entre personajes. Al principio se consideró que, si se avanzaba de capítulo, se seguía en el mismo contexto que en el capítulo anterior.

La solución alcanzada para esta nueva funcionalidad implicó un esfuerzo adicional que si se hubiese tenido en cuenta desde el principio no habría supuesto.

Las conclusiones que podemos obtener son las siguientes, al igual que en la formación, dedicar un esfuerzo extra a la especificación de requisitos, escribiéndolos y realizando preguntas al cliente, reducirá los problemas que surgen si no se tienen completamente claros los requisitos. También es importante tener en cuenta un posible

aumento de requisitos a mitad del desarrollo, puede hacer que los cambios a realizar no tengan un gran impacto en el coste de tiempo.

5.4. Algoritmo de creación automática de diccionario

Para generar el diccionario se realizaron varias pruebas para comprobar cuál era el método más efectivo. Partimos de la suposición de que las palabras en mayúsculas se corresponden con nombres de personajes.

- En la primera aproximación se considera que, si hay dos o más palabras en mayúsculas tras un signo de puntuación, esta combinación se corresponde con un nombre.
- En la segunda aproximación se considera que la primera palabra después de un signo de puntuación nunca es un nombre a pesar de que esté acompañado por otras palabras en mayúsculas.

Tras comparar los resultados de ambas aproximaciones se determina lo siguiente:

- Ambas aproximaciones generan falsos positivos. Esto se debe principalmente a lugares, ríos, mares etc. y a recursos narrativos que puede utilizar el autor y que la aproximación no realiza bien.
- La segunda aproximación reduce el número de falsos positivos considerablemente.
- La segunda opción también genera resultados contra intuitivos y genera nombres que al buscar las posiciones aparecen con 0 apariciones. Un ejemplo de estos casos es cuando un nombre compuesto siempre aparece como tal, por ejemplo, Jon Snow siempre aparece como Jon Snow; en el caso de que en una de esas apariciones ocurriese tras un signo de puntuación se añadiría Snow como nombre. Si Snow siempre viene precedido de Jon, nunca se considerará al ser Jon Snow una cadena de caracteres más larga que Snow.

5.5. Algoritmo de obtención de posición de personajes

Los personajes del diccionario son un conjunto de datos dinámicos, por lo que no sabemos qué nombres vamos a tener que encontrar a la hora buscar posiciones.

Ply solo permite especificar los patrones de los tokens estáticamente. En algún foro se sugiere la opción de introducir la expresión regular que contenga el dato dinámico a través de un decorador que ofrece la librería, pero tras probarlo se puede afirmar que esta solución no funciona.

Se toma la decisión de generar un analizador léxico muy genérico, para encontrar todos los posibles nombres que pueda haber. Si empieza por cualquier carácter salvo un signo de puntuación se considera una palabra y se realizan las acciones pertinentes

para detectar si se corresponde o puede llegar a corresponderse con un nombre del diccionario o no. (Más explicaciones en la guía de programador).

5.6. Integración de network_styling_with_d3

A la hora de representar la red, se encontró una librería llamada Netwulf que permitía hacer unas representaciones muy visuales y personalizables. Lo malo de esta librería era que no se podía modificar la página web que genera para la visualización de una forma sencilla. Leyendo detenidamente la documentación del repositorio se encontró la aplicación web que utiliza Netwulf para su visualización, por lo que se procedió a descargar e integrar la aplicación web en nuestra página web.

Durante la integración de la aplicación web surgieron dos problemas:

- Cacheo de los ficheros JS por parte del navegador.
- Grandes cantidades de código y pocos comentarios.

El primer problema fue debido a mi inexperiencia a la hora de desarrollar webs. El problema surgía del cacheo que hacen los navegadores a ficheros tanto JS como CSS. Este es un problema muy fácil de solucionar si se sabe que esto es algo que ocurre y un verdadero quebradero de cabeza si ves cómo modificas el código y tus cambios no se ven reflejados. La solución más sencilla para forzar la actualización de estos ficheros en el navegador es pulsar Ctrl+F5.

El segundo problema vino por la falta de comentarios de la aplicación web. Esto supuso un problema debido a que se querían hacer las modificaciones oportunas para guardar la configuración visual que le había dado el usuario a la red, y para visualizar la red generada por nuestra aplicación; debido a la falta de comentarios explicativos sobre las distintas funcionalidades del código y la gran cantidad de líneas de código que tiene esta aplicación las modificaciones acabaron siendo una ardua tarea.

La conclusión a la que he podido llegar, tras integrar esta aplicación en la nuestra, es que como bien nos comentan los profesores desde que empezamos el grado, comentar el código es muy importante. Comentar el código ya no solo te facilita el trabajo cuando necesitas hacer modificaciones sobre la función, sino que, además, facilita a terceros entender el código.

5.7. Arquitectura MVP

El motivo principal para usar esta arquitectura es aislar la lógica del proyecto de la interfaz para así conseguir un código más fácil de mantener, de forma que, si se modifica la lógica del programa, no haga falta modificar también la interfaz.

En el proyecto la vista está formada por el conjunto de plantillas HTML, que notifican al controlador (presentador) las acciones del usuario; Flask, que actúa como controlador, le manda estas actualizaciones al modelo, que se encarga de realizar las

acciones oportunas. El modelo notifica al controlador los cambios y este informa de las actualizaciones a la vista.

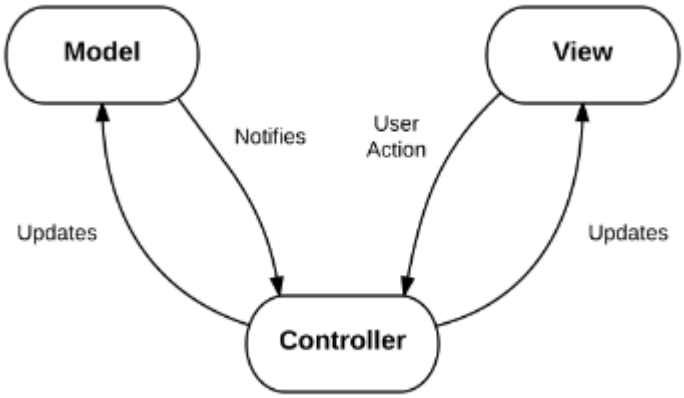

Figura E 1 Esquema Arquitectura MVC (44)

5.8. Detección de roles

Se realiza una implementación propia del algoritmo de detección de roles, que actúa sobre comunidades ya generadas por algoritmos de NetworkX.

Esta implementación supone una mejora en la eficiencia de algoritmos previos de detección de roles, que utilizaban recocidos simulados (12) para calcular la partición con máxima modularidad. En nuestra implementación, el algoritmo se basa en el algoritmo de Girvan-Newman (12).

F. Trabajos relacionados

A lo largo de los años se han realizado varios estudios sobre las relaciones entre personajes en distintas novelas intentando extraer diversos tipos de información. Algunas como la red de tronos (45) se acerca más a la aproximación seguida en nuestra aplicación y otras como Extracting Social Network from Literature to Predict Antagonist and Protagonist o Extracting Social Networks from Literary Fiction (46,47) sin embargo utilizan técnicas de minería de datos o redes neuronales para obtener sus datos.

A continuación, se exponen algunos de los estudios realizados mostrando sus puntos fuertes y débiles a la hora de tratar de obtener información de las novelas.

6.1. Network of Thrones

Un proyecto similar al que estamos realizando es el de Network of Thrones, creado por Andrew Beveridge y Jie Shan (45). Este proyecto realizó una serie de pasos para generar la red:

- Lo primero que hicieron fue parsear los libros en busca de palabras en mayúscula y también utilizaron el web scraping sobre una wiki especializada, para obtener los nombres de los personajes.
- Después trataron de quitar las ambigüedades que pudiera haber. Un ejemplo de una ambigüedad que encontraron fue entre Jon Arryn y Jon Snow; ambos personajes comparten el primer nombre, lo que genera una ambigüedad cuando aparece únicamente Jon en el texto.
- Para solucionar este problema lo que hicieron fue reescribir el libro para eliminar las ambigüedades. Cada vez que encontraban una ambigüedad, la sustituían por un identificador definido por ellos; por ejemplo, sustituían “king” por “king_Joffrey” cuando la palabra rey se refería a este monarca.
- Para generar los enlaces entre los personajes, trataron de detectar distintas interacciones mediante una cercanía de aparición de palabras. Las relaciones que detectan son las siguientes:
    - Que dos personajes aparezcan en la misma ubicación.
    - Que haya una conversación entre dos personajes.
    - Que un personaje esté hablando sobre otro personaje.
    - Que un personaje escuche a un tercer personaje hablando sobre un segundo.
    - Que un tercer personaje hable sobre otros dos personajes.
    - Y otras formas más de interacción.
- Finalmente crearon la red a partir de una distancia de 15 palabras al considerar esta distancia la mejor para los libros.

6.2. Extracting Social Network from Literature to Predict Antagonist and Protagonist.

En este estudio (46) lo que se busca es encontrar los protagonistas y antagonistas de diversas novelas. Su algoritmo se basa en cuatro puntos que son los siguientes:

- Primero eliminaron información irrelevante de los textos como son las imágenes o los títulos de capítulo.
- Después, identificaron los personajes del libro.
- Más tarde, identificaron relaciones entre personajes y si estas son relaciones positivas o negativas, dando un peso de 1 a las relaciones positivas y de -0.5 a las negativas.
- Finalmente, utilizaron los datos resultantes para realizar distintos análisis sobre la red social generada, además de determinar quién es el personaje protagonista y cual el antagonista.

A la hora de identificar personajes del libro se encontraron con diversos problemas con las palabras que indirectamente se refieren a personajes, como pueden ser “ellos”, “vosotros”, etc. Para solucionar este problema asignaron estas palabras al personaje más cercano. Otro problema que encontraron fue la incorrecta detección de personajes que tuvieron que solucionar eliminando la referencia al personaje o reasignándola.

Para determinar si la relación entre personajes era positiva o negativa utilizaron el diccionario SentiWordNet (46).

6.3. Extracting Social Networks from Literary Fiction

En este trabajo (47) lo que se busca es extraer redes de conversaciones entre personajes. En esta red cada enlace muestra al menos un dialogo entre personajes, y definen una conversación entre personajes mediante una serie de condiciones:

- Los personajes se encuentran en el mismo lugar.
- Los personajes se turnan para hablar.
- Los personajes son conscientes del otro personaje y pretenden que lo que dicen sea escuchado por el otro personaje.

Como en los otros ejemplos, también siguen una serie de pasos para generar los datos:

- Lo primero que hacen es preprocesar el texto, para normalizar el formato, detectar capítulos, eliminar metadatos e identificar diálogos.
- Después tratan de detectar personajes. Esto lo hacen con el Stanford NER tagger, que es un clasificador que sirve para detectar nombres y

organizaciones (48), y después generan variaciones del propio nombre, como pueden ser Sherlock Holmes y Mr. Holmes.

- Tras esto, lo que hacen es asignar personajes a cada dialogo mediante técnicas de minería de datos.
- Para construir la red filtraron los personajes que aparecían menos de 3 veces o generaban menos del 1% de las menciones del libro al considerarlas como ruidosas, y de esta forma obtener mejores resultados.

G. Conclusiones y líneas de trabajo futuras

En esta sección se redactan las conclusiones obtenidas tras el desarrollo del proyecto, y se proponen nuevas líneas de trabajo futuro.

7.1. Conclusiones

Al finalizar el proyecto se cumplen los objetivos marcados al inicio: se permite generar una red de interacción de personajes de cualquier EPUB, y se genera un informe donde se analizan las métricas más importantes de la red.

Se han alcanzado los siguientes objetivos personales con el desarrollo del proyecto:

- Se han aumentado los conocimientos de desarrollo web, ya sea en HTML, como en JS y CSS.
- Se ha vivido una experiencia similar a la que sería el desarrollo de un proyecto en una empresa utilizando metodología ágil.
- Se ha visto la utilidad de lo aprendido en diversas asignaturas cursadas, como pueden ser Procesadores del Lenguaje, debido a la utilización de un analizador léxico, Sistemas Distribuidos etc.

Con la finalización del proyecto se alcanzan las finalidades propuestas en la introducción.

Se proporciona una aplicación desde la que cualquier usuario puede generar una red compleja acorde a sus intereses personales y se le permite tanto la visualización de la red como la obtención de métricas interesantes. Esto lo hace de una manera sencilla y sin que el usuario deba ser un experto en redes complejas ni para generar la propia red ni para extraer la información relevante.

También se permite la extracción de propiedades de las novelas para ser utilizadas en sistemas de recomendación. Para ello se da la opción de exportar la red generada a distintos formatos. Con la red exportada se podría utilizar otra herramienta que analice las métricas que considere oportunas y las utilice para recomendar otras novelas.

7.2. Líneas de trabajo futuras

Durante el desarrollo del proyecto se han ido sugiriendo funcionalidades adicionales que mejorarían en gran medida la aplicación, pero que por cuestiones de tiempo no se han podido implementar o se podrían haber implementado de una mejor forma.

Internacionalización de network_styling_with_d3

Como hemos comentado en la sección de aspectos relevantes, integrar esta aplicación web fue arduo debido a los pocos comentarios y la gran cantidad de líneas de código. A la hora de internacionalizar la aplicación, también se buscó la internacionalización del menú de personalización que ofrece la aplicación. Tras diversas pruebas se comprobó que con la implementación actual de la aplicación no se podía añadir un idioma sin tener que tener una variable nueva de configuración de la red, por lo que se decidió posponer la internacionalización de esta parte.

Posibles opciones:

- Tener una variable de configuración por cada idioma y actualizar todas cuando se actualice una de ellas.
- Al tener la app una licencia MIT, modificar el código oportuno para que se pueda guardar solo una variable de configuración sin verse influida por el idioma en que se muestra la página.

Resolución de ambigüedades

Al igual que en otros trabajos similares, para la generación de redes de interacción se tiene que lidiar con ambigüedades a la hora de reconocer las posiciones de todos los nombres en el texto.

Las opciones que se podrían implementar para solucionar este problema y que no son necesariamente exclusivas son las siguientes:

- Asistente para el usuario. Podemos dejar en manos del usuario la resolución de ambigüedades mostrándole el texto donde se encuentre alguna de ellas y que el propio usuario las resuelva por el contexto.
- Realizar predicciones de a quién se refiere la ambigüedad por los personajes en el radio de interacción con esa ambigüedad. Si un personaje está muy relacionado con otro, y el primero aparece en el radio de interacción de la ambigüedad, es probable que el segundo sea el productor de la ambigüedad.

Mejora del sistema de sesión de usuario

Actualmente para no guardar información de usuarios innecesaria se comprueba cuándo el usuario abandona la página. El problema es que esta forma de comprobarlo no es completamente fiable, cuando el usuario utiliza las flechas de avanzar, retroceder o actualizar se detecta que el usuario ha abandonado la página y se borran todos los datos pertenecientes a él.

Las posibles opciones para que esto no ocurra son:

- Modificar el código JavaScript de los HTML para que no detecte estas interacciones como un abandono de la web.
- Sustituir el sistema actual por un sistema en el que cuando un usuario pase más de un determinado tiempo sin interactuar con la página se elimine su información.

Otro problema derivado de este sistema es que actualmente, puede dejar de funcionar aun sin modificar el código. El motivo de estos problemas no se ha investigado en profundidad por falta de tiempo, pero se presupone que es debido a actualizaciones de los navegadores, JavaScript o jQuery.

Nuevas medidas de análisis

Por cuestiones de tiempo, se han implementado una serie de medidas de análisis que pueden quedarse cortas para usuarios más avanzados. Una línea de trabajo futuro sería aumentar el número de medidas de análisis para que el usuario pueda obtener mejor información sobre la red de interacción generada.

Avisos a los usuarios

Actualmente se le otorga información al usuario mediante alerts de JavaScript. Aunque este método funciona, una posible mejora sería eliminar los alerts y sustituirlos por mensajes dentro del HTML de la propia página que serían solo visibles cuando ocurra el correspondiente evento.

Bibliografía

1. Ciencia de redes. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 11 de junio de 2019]. Disponible en: https://es.wikipedia.org/w/index.php?title=Ciencia_de_redes&oldid=116172205

2. How Social Networking Helped Capture Saddam [Internet]. NPR.org. [citado 11 de junio de 2019]. Disponible en: https://www.npr.org/templates/story/story.php?storyId=124052190

3. Understanding Digital Humanities | D. Berry | Palgrave Macmillan [Internet]. [citado 24 de junio de 2019]. Disponible en: https://www.palgrave.com/gp/book/9780230292642

4. Linden G, Smith B, York J. Linden G, Smith B and York J: 'Amazon.com recommendations: item-to-item collaborative filtering', Internet Comput. IEEE, , 7. Internet Computing, IEEE. 1 de febrero de 2003;7:76-80.

5. EPUB. En: Wikipedia [Internet]. 2019 [citado 18 de enero de 2019]. Disponible en: https://en.wikipedia.org/w/index.php?title=EPUB&oldid=878221726

6. EPUB 3 Overview [Internet]. [citado 18 de enero de 2019]. Disponible en: http://www.idpf.org/epub/30/spec/epub30-overview.html#sec-nav

7. Network Science (Camb02): Amazon.es: Albert-László Barabási, Márton Pósfai: Amazon.es [Internet]. [citado 12 de junio de 2019]. Disponible en: https://www.amazon.es/Network-Science-Camb02-Albert-L%C3%A1szl%C3%B3-Barab%C3%A1si/dp/1107076269/ref=sr_1_1?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&keywords=barabasi+networks&qid=1560370665&s=gateway&sr=8-1

8. Networks: Amazon.es: Mark Newman: Amazon.es [Internet]. [citado 12 de junio de 2019]. Disponible en: https://www.amazon.es/Networks-Mark-Newman/dp/0198805098/ref=sr_1_1?__mk_es_ES=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=3B3FKIEWN0HSY&keywords=newman+networks&qid=1560370643&s=gateway&sprefix=newman+%2Caps%2C151&sr=8-1

9. Lancichinetti A, Fortunato S. Community detection algorithms: A comparative analysis. Phys Rev E [Internet]. 30 de noviembre de 2009 [citado 24 de junio de 2019];80(5):056117. Disponible en: https://link.aps.org/doi/10.1103/PhysRevE.80.056117

10. Fortunato S. Community detection in graphs. Physics Reports [Internet]. febrero de 2010 [citado 24 de junio de 2019];486(3-5):75-174. Disponible en: https://linkinghub.elsevier.com/retrieve/pii/S0370157309002841

11. Newman MEJ. Modularity and community structure in networks. Proc Natl Acad Sci USA. 6 de junio de 2006;103(23):8577-82.

12. Cartography of complex networks: modules and universal roles [Internet]. [citado 12 de junio de 2019]. Disponible en: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2151742/

13. Lexical analysis. En: Wikipedia [Internet]. 2019 [citado 9 de junio de 2019]. Disponible en: https://en.wikipedia.org/w/index.php?title=Lexical_analysis&oldid=899822734

14. Cesar Ignacio García Osorio. Apuntes de la asignatura Procesadores del Lenguaje. 2017.

15. Tchirou F. Lexical Analysis [Internet]. Hacker Noon. 2017 [citado 9 de junio de 2019]. Disponible en: https://hackernoon.com/lexical-analysis-861b8bfe4cb0

16. Web scraping. En: Wikipedia, la enciclopedia libre [Internet]. 2018 [citado 9 de junio de 2019]. Disponible en: https://es.wikipedia.org/w/index.php?title=Web_scraping&oldid=112435576

17. Qué es el Web scraping? Introducción y herramientas [Internet]. Sitelabs. 2016 [citado 9 de junio de 2019]. Disponible en: https://sitelabs.es/web-scraping-introduccion-y-herramientas/

18. Scrum (desarrollo de software) - Wikipedia, la enciclopedia libre [Internet]. [citado 10 de junio de 2019]. Disponible en: https://es.wikipedia.org/wiki/Scrum_(desarrollo_de_software)

19. Git. En: Wikipedia, la enciclopedia libre [Internet]. 2019 [citado 16 de mayo de 2019]. Disponible en: https://es.wikipedia.org/w/index.php?title=Git&oldid=115198516

20. Hello World · GitHub Guides [Internet]. [citado 10 de junio de 2019]. Disponible en: https://guides.github.com/activities/hello-world/

21. FAQs and Support - Your GitHub Issue Tracker [Internet]. [citado 10 de junio de 2019]. Disponible en: https://www.zenhub.com/faq

22. word processor integration [Zotero Documentation] [Internet]. [citado 15 de enero de 2019]. Disponible en: https://www.zotero.org/support/word_processor_integration

23. Zotero | Downloads [Internet]. [citado 15 de enero de 2019]. Disponible en: https://www.zotero.org/download/

24. Cite Websites with a Browser Plugin - Mendeley Web Importer [Internet]. [citado 15 de enero de 2019]. Disponible en: https://www.mendeley.com/reference-management/web-importer

25. Bibliographic/Software and Standards Information - Apache OpenOffice Wiki [Internet]. [citado 15 de enero de 2019]. Disponible en: https://wiki.openoffice.org/wiki/Bibliographic/Software_and_Standards_Information#Mendeley

26. Home - Pencil Project [Internet]. [citado 15 de enero de 2019]. Disponible en: https://pencil.evolus.vn/

27. Features - Pencil Project [Internet]. [citado 15 de enero de 2019]. Disponible en: https://pencil.evolus.vn/Features.html

28. Download free Adobe XD CC | UX/UI design and collaboration tool [Internet]. [citado 15 de enero de 2019]. Disponible en: https://www.adobe.com/es/products/xd.html

29. 13.5. zipfile — Work with ZIP archives — Python 3.6.8 documentation [Internet]. [citado 21 de enero de 2019]. Disponible en: https://docs.python.org/3.6/library/zipfile.html

30. 10.10. shutil — High-level file operations — Python 2.7.16 documentation [Internet]. [citado 10 de junio de 2019]. Disponible en: https://docs.python.org/2/library/shutil.html

31. secrets — Generate secure random numbers for managing secrets — Python 3.7.4rc1 documentation [Internet]. [citado 25 de junio de 2019]. Disponible en: https://docs.python.org/3/library/secrets.html

32. Overview of NetworkX — NetworkX 2.3 documentation [Internet]. [citado 10 de junio de 2019]. Disponible en: https://networkx.github.io/documentation/stable/

33. ulfaslak/network_styling_with_d3: (1) Input a network. (2) Style it. (3) Download the result. [Internet]. [citado 10 de junio de 2019]. Disponible en: https://github.com/ulfaslak/network_styling_with_d3

34. benmaier/netwulf: Interactive visualization of networks based on Ulf Aslak's d3 web app. [Internet]. [citado 10 de junio de 2019]. Disponible en: https://github.com/benmaier/netwulf

35. Beautiful Soup Documentation — Beautiful Soup 4.4.0 documentation [Internet]. [citado 21 de enero de 2019]. Disponible en: https://www.crummy.com/software/BeautifulSoup/bs4/doc/index.html

36. 21.5. urllib — URL handling modules — Python 3.6.8 documentation [Internet]. [citado 10 de junio de 2019]. Disponible en: https://docs.python.org/3.6/library/urllib.html

37. PLY (Python Lex-Yacc) [Internet]. [citado 27 de enero de 2019]. Disponible en: http://www.dabeaz.com/ply/

38. Welcome to Flask — Flask 1.0.2 documentation [Internet]. [citado 10 de junio de 2019]. Disponible en: http://flask.pocoo.org/docs/1.0/

39. The Flask Mega-Tutorial Part I: Hello, World! - miguelgrinberg.com [Internet]. [citado 10 de junio de 2019]. Disponible en: https://blog.miguelgrinberg.com/post/the-flask-mega-tutorial-part-i-hello-world

40. Flask-Babel — Flask Babel 1.0 documentation [Internet]. [citado 10 de junio de 2019]. Disponible en: https://pythonhosted.org/Flask-Babel/

41. JavaScript - Wikipedia, la enciclopedia libre [Internet]. [citado 10 de junio de 2019]. Disponible en: https://es.wikipedia.org/wiki/JavaScript

42. jQuery - Wikipedia, la enciclopedia libre [Internet]. [citado 10 de junio de 2019]. Disponible en: https://es.wikipedia.org/wiki/JQuery

43. sentdex. Flask Tutorial Web Development with Python 1 - Intro [Internet]. [citado 11 de junio de 2019]. Disponible en: https://www.youtube.com/watch?v=Lv1fv-HmkQo&list=PLQVvvaa0QuDc_owjTbIY4rbgXOFkUYOUB

44. Sivji A. Building a Flask Web Application (Flask Part 2) [Internet]. [citado 11 de junio de 2019]. Disponible en: ./flask-part2-building-a-flask-web-application.html

45. From Book to Network [Internet]. Network of Thrones. 2017 [citado 16 de enero de 2019]. Disponible en: https://networkofthrones.wordpress.com/from-book-to-network/

46. Michael Peterson, Matt Fernandez, Ben Ulmer. Extracting Social Network from Literature to Predict Antagonist and Protagonist. 7 de diciembre de 2015; Disponible en: https://nlp.stanford.edu/courses/cs224n/2015/reports/14.pdf

47. Extracting social networks from literary fiction [Internet]. [citado 10 de junio de 2019]. Disponible en: https://dl.acm.org/citation.cfm?id=1858696

48. The Stanford Natural Language Processing Group [Internet]. [citado 10 de junio de 2019]. Disponible en: https://nlp.stanford.edu/software/CRF-NER.html