Evolución temporal en espacios semánticos

Ariel Agustín Berardino

Tesis de Licenciatura en Ciencias Físicas

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Marzo 2020

- TEMA: Evolución temporal en espacios semánticos.
- ALUMNO: Ariel Agustín Berardino.
- LU: 327/13.
- DIRECTOR DEL TRABAJO: Enzo Tagliazucchi.
- FECHA DE INICIACIÓN: Abril 2019.
- FECHA DE FINALIZACIÓN: Marzo 2020.
- FECHA DE EXAMEN:
- INFORME FINAL APROBADO POR:

Autor	Jurado
Director	Jurado
Profesor de Tesis de Licenciatura	Jurado

"Love is the one thing we're capable of perceiving that transcends dimensions of time and space. Maybe we should trust that, even if we can't understand it."

Brand to Cooper from Interstellar

UNIVERSIDAD DE BUENOS AIRES

Resumen

Facultad de Ciencias Exactas y Naturales

Departamento de Física

Licenciatura en Ciencias Físicas

Hay distintas características de las formas de arte que evolucionan en períodos bien definidos, y dan cuenta de variables intrínsecas a cada una de ellas, tales como el espectro rítmico musical (Levitin et al., 2012), o el uso de ciertas formas lingüísticas en libros (Michel et al., 2011); así como de variables espontáneas en el proceso de evolución, por ejemplo, la búsqueda deliberada de nuevas formas de expresión artística. En esta tesis proponemos un método para evaluar en un conjunto de documentos si hay un efecto intrínseco asociado al orden temporal de su publicación, y lo ilustramos en el caso de subtítulos de películas. En particular estudiamos el contenido semántico del texto de los archivos de subtítulos, detectando tópicos semánticos dentro del corpus con el método no supervisado: factorización no negativa de matrices (NMF). Luego construimos el grafo semántico cuya matriz de adyacencia fue computada mediante la correlación de Pearson de los vectores de la matriz W de "documentos por tópicos", previamente generada con el método NMF para la detección de los tópicos. Por último, construimos caminos ordenados y aleatorios para recorrer el grafo pasando por todos sus nodos y calculamos su longitud, definida como la suma de los pesos de los enlaces entre nodos consecutivos de los caminos. Al recorrer el grafo en orden cronológico obtuvimos caminos más pesados que al permutar el orden de los nodos de manera al azar. También notamos que recorriendo el grafo por tópicos, y dentro de los tópicos por años, se obtuvieron caminos más largos que al atravesarlo sólo por tópicos. De esta manera podemos inferir una correlación a escala temporal del contenido semántico en el texto de subtítulos de películas.

Agradecimientos

Este trabajo no hubiera sido posible sin la ayuda de muchas personas que de forma directa o indirecta contribuyeron a su realización. Por ello quería dedicarles unas palabras de agradecimiento:

A Enzo por guiarme en este proceso de culminación de la carrera de grado con un trabajo de investigación que desde el primer momento me pareció súper interesante. También por su predisposición constante, transmisión de conocimiento, buena onda, hiperactividad y por bancarme todo este tiempo.

A Aye por ser la persona más increíble que tuve el enorme placer de conocer y compartir la mayor parte de estos dos últimos años. Su apoyo y consejos son los ingredientes indispensables de mi buen comportamiento.

A Lili y Cacho porque sin ellos no estaría acá, literalmente. Pero sobretodo por el amor y cuidado con los que me criaron y por darme todos los gustos.

A la Bach por mantener brillando su luz interior a pesar de ser la persona que más se bancó mis jodas e insoportabilidad. A Bruno por ser un ejemplo a seguir en todos los aspectos y ayudarme desde donde sea.

A la abuela por sus milanesas napolitanas y ensalada rusa que son siempre una excusa para juntarnos con la familia. Al tío por ser la persona más buena del planeta y a Javi por ser un hermano y un amigo más que un primo.

A 'le park', que son mi familia por elección, por todos los asados habidos y por haber. PD: Sí era yogurt. A los y las humeantes que lograron que el estudio de esta licenciatura fuera mucho más ameno y placentero. A la liviana por tratar siempre de ser más pesada pero fallar enormemente en el intento. Al fútbol dominguero por ser la actividad con más constancia de la historia hasta la fecha, y permitirme descargar todo en esos 60 minutos.

A la familia y amigues que no nombré hasta ahora pero que son una parte muy importante en mi vida: mi madrina, Pablo, Miri, Joaqui, Rami, Andi, Maru, Bella, Flori, Luis, Panchito, Fer, Soli, Mica, Enrique, Luciana, Santi, Carlitos, Marité, Marina, Vale, Lili, Dani, Ailu y más personas que seguramente esté olvidando mencionar en estas líneas.

GRACIAS TOTALES

Índice general

Capítulo 1

Introducción

El análisis de señales empíricas es la antesala de la formulación de los modelos matemáticos que se encuentran en el núcleo de las ciencias físicas. Sin embargo, la mayoría de los datos disponibles en la actualidad no se amoldan inmediatamente a este modelo, sino que se los considera datos complejos porque no pueden ser reducidos de forma natural a una secuencia numérica (Van Der Aalst, 2016). Ejemplos de datos complejos incluyen secuencias simbólicas (como lenguaje natural, o secuencias de ADN), matrices de valores de alta dimensionalidad (como por ejemplo imágenes o video), o datos relacionados entre sí de forma compleja (por ejemplo, redes o grafos) (Van Der Aalst, 2016; Goldberg, 2016). Este tipo de datos son de gran interés en la actualidad por representar la enorme mayoría de datos generados por la sociedad y economía global y disponibles en la WWW. La formulación de modelos probabilísticos para comprender la evolución temporal de datos complejos es un requisito para organizar y sistematizar los datos y transformarlos en información útil.

El estudio de la evolución temporal de datos complejos es relativamente reciente debido a las demandas computacionales de trabajar con una representación completa y esparsa de los datos. En cambio, la aplicación de embeddings (Goldberg, 2016) y modelos de aprendizaje no supervisado (Alghamdi and Alfalqi, 2015) para la reducción de la dimensionalidad permite asociar a cada lapso temporal un vector no esparso de tamaño más reducido, facilitando el estudio y modelado matemático de su evolución temporal.

Existen trabajos que estudian datos complejos y que muestran distintas características de las formas de arte que evolucionan en períodos bien definidos, y dan cuenta de variables intrínsecas a cada una de ellas, así como de variables espontáneas en el proceso de evolución, por ejemplo, la búsqueda deliberada de nuevas formas de expresión artística. Mencionando un caso particular, se ha estudiado la evolución histórica de la predictibilidad en el espectro rítmico musical y cómo ésta favorece el disfrute de la pieza

(Levitin et al., 2012). Otro ejemplo es el estudio de cómo fue evolucionando temporalmente el uso de ciertas formas lingüísticas en libros en inglés, en particular, el fenómeno de *regularización* de verbos irregulares (y viceversa), y cómo éste es afectado por la incorporación de palabras en el vocabulario (Michel et al., 2011). También podemos encontrar un estudio de las 'trayectorias' (o arcos) narrativos, donde se reporta que el contenido emotivo, *sentiment analysis*, dentro de las historias de distintos autores está dominado por seis formas básicas (Reagan et al., 2016). Finalmente, un estudio masivo ha evaluado el uso de colores, texturas y formas artísticas en pintura a lo largo del tiempo, y cómo estas variables dependen de factores coyunturales como la disponibilidad de ciertos pigmentos (Kim et al., 2014).

En la presente tesis evaluamos la presencia de marcadores que indiquen un efecto cronológico en la evolución de contenido semántico; es decir, buscamos entender si la similitud semántica en un corpus dado manifiesta un efecto de continuidad histórica, o refleja únicamente la división entre determinados géneros. Para esto consideramos un corpus más acotado que el estudiado por Michel et al. (2011), pero lo suficientemente amplio y extendido en el tiempo como para que sea interesante analizarlo desde esta perspectiva: el conjunto de *todos* los subtítulos disponibles en la base de datos de Open Subtitles (`https://www.opensubtitles.org/en/search/subs`), el cual comprende la mayoría de obras cinematográficas producidas durante este siglo y el anterior.

Proponemos, entonces, extender en esta tesis los trabajos anteriores en múltiples direcciones:

- En primer lugar, buscamos capturar la evolución del contenido semántico en un sentido amplio (no únicamente el contenido emotivo) mediante la aplicación de técnicas de clusterización de la matriz de co-ocurrencia de palabras.
- En segundo lugar, buscamos introducir un método para representar mediante nodos los momentos determinados en la evolución temporal y en sus conexiones la similitud semántica del contenido asociado a los nodos. Recorrer dichos nodos en el orden temporal preferido representa un costo asociado a la sumatoria de los pesos de las conexiones, el cual puede ser comparado con la distribución de pesos obtenida mediante un ensemble de múltiples caminos que recorren la red en sentidos no-cronológicos (en esta tesis los caminos son escogidos al azar, aunque este método puede adaptarse para recorrer la red con más restricciones).

Para desarrollar e investigar estas herramientas proponemos analizar un gran corpus de subtítulos de aproximadamente 170.000 películas (en inglés y en formato .srt). La idea es utilizar este tipo de datos dado que son sencillos de obtener, de asignarle tópicos

bien definidos (Rabinovich and Girdhar, 2015; Mocanu et al., 2016; Bougiatiotis and Giannakopoulos, 2017) y también dado que el texto correspondiente contiene explícitamente una etiqueta del tiempo transcurrido desde el comienzo. Esto es una ventaja con respecto a otros corpus de texto como novelas, cuentos, revistas, diarios, blogs, o *tweets* que carecen de un posible etiquetado temporal consistente para cada palabra del texto, y da lugar a investigar la evolución del contenido semántico **dentro** de la obra misma (un análisis que discutimos como una posible extensión de esta tesis en el capítulo 7 de conclusiones).

Esta tesis está estructurada de la siguiente manera. Los dos próximos capítulos contienen una presentación sucinta de la metodología de procesamiento de lenguaje natural requerida para el análisis, así como una introducción a conceptos básicos de redes. El capítulo siguiente contiene una caracterización estadística de la base de datos. Los últimos dos capítulos de la tesis contienen la aplicación del método NMF para la detección de tópicos a lo largo del tiempo, y su uso para definir una red de similitud de contenido semántico, la cual es recorrida de diversas formas para entender el efecto del orden cronológico en la similitud semántica de los subtítulos.

Capítulo 2

Métodos para el procesamiento de textos

2.1. Introducción metodológica

En este capítulo explicaremos las herramientas de análisis semántico que utilizamos para realizar la detección de tópicos en los textos generados a partir de subtítulos de películas. A su vez, contaremos cómo representar un texto en forma matricial y simbólica, analizando un ejemplo sencillo de NLP[1].

El primer paso en el análisis consta de remover símbolos, números, no-palabras, palabras que no están en el diccionario inglés de la librería de Python *Natural Lenguage ToolKit* (NLTK: `https://www.nltk.org/`) (Loper and Bird, 2002). Luego, expandir las contracciones propias del idioma, para después tokenizar, determinar el rol gramatical de las palabras en la oración (POS tag), lematizar, y calcular la matriz TF-IDF[2] que representa al texto en base a las palabras en el vocabulario y la frecuencia con la que aparecen en el texto. Cada uno de estos pasos se detallará en la sección 2.1.2. Por último, contaremos qué es el modelo *non-negative matrix factorization* (NMF) (Xu et al., 2003) que utilizamos para la detección de tópicos en los datos y veremos ejemplos para ilustrar los procedimiento en cada caso. Todos los scripts usados para el procesamiento del texto fueron desarrollados en Python 3.1.

[1] Natural language Processing
[2] Term frequency – Inverse document frequency

2.1.1. Procesamiento de lenguaje natural

El procesamiento del lenguaje natural (NLP por sus siglas en inglés) consiste en desarrollar aplicaciones y servicios que puedan comprender y procesar lenguaje producido por humanos. Algunos ejemplos prácticos son el reconocimiento de voz, la traducción del habla, la comprensión de oraciones completas, la comprensión de sinónimos de palabras coincidentes y la escritura de oraciones y párrafos completos gramaticalmente correctos (Collobert et al., 2011). Hay numerosas librerías de NLP de código abierto, en esta tesis usamos *Natural Language ToolKit (NLTK)* (Loper and Bird, 2002). La librería NLTK contiene los métodos necesarios para procesar texto, detectar y visualizar tópicos, los cuales son descriptos a continuación.

2.1.2. Preprocesamiento del texto

Las bases de datos utilizadas en la tesis están conformadas por los subtítulos de películas creados por los usuarios del sitio web *opensubtitles.org*, cada uno de ellos en formato de archivos de texto *.srt*. Para obtener una representación matricial del texto se implementó una serie de pasos de **pre-procesamiento**, y así luego poder realizar el análisis semántico de los datos.

Esta primera etapa consistió en los pasos que enumeramos a continuación:

- **Remoción de símbolos**. Eliminamos los símbolos de puntuación, exclamación, interrogación, numéricos, otros símbolos y marcas de hipertexto del lenguaje. e.g. "<body> We're good people. We aint bad < \body>" por "We're good people We aint bad"
- **Minúsculas**. Cambiamos todas las mayúsculas del texto por minúsculas.
- **Expansión de contracciones típicas del idioma**. Para facilitar el siguiente paso compilamos una lista de las contracciones del idioma inglés más usadas en el lenguaje formal e informal y las reemplazamos por su expansión correspondiente, e.g. "we**'re** good people we **aint** bad" por "we **are** good people we **are not** bad"
- **Tokenización**. Dividir texto en *tokens*. El tipo de tokens depende de las reglas usadas para la división del texto. Estos pueden ser oraciones si dividimos en oraciones el texto de un párrafo, o pueden ser palabras si dividimos en palabras el texto de cada oración. Usando la función **word_tokenize()** de la biblioteca *NLTK* tokenizamos cada línea de subtítulos cortándola por espacios en blanco o signos de puntuación, exclamación, etc. Por ejemplo, al tokenizar la línea de subtítulos

"Hey, I'm walkin' here!" obtenemos la lista de tokens = ["Hey", ",", "I", "'m", "walkin", "'", "here", "!"].

- **POS tagging**. Identificamos el rol gramatical de cada palabra (*part of speech tagging*)(Schmid, 1995) con la función **pos_tag()** incluida en la biblioteca NLTK, es decir, determinamos si la palabra es un verbo, sustantivo, adjetivo, etc.

- **Lematización**. Con los tokens y el POS tagging pudimos encontrar el lema o raíz de cada palabra y reemplazamos todas las palabras por su lema correspondiente, e.g. "playing" por "play", "trees" por "tree", etc.

- **Remoción de palabras "indeseadas"**. Removimos las palabras que no aparecían en el diccionario de idioma inglés de NLTK y en su lista de *stopwords*, es decir, de palabras que no poseen significado semántico bien definido, tales como artículos, pronombres, preposiciones, etc.

Al terminar con el pre-procesado obtuvimos para cada archivo una lista con todas las palabras lematizadas utilizadas en cada uno de ellos. A continuación, representamos al corpus como una matriz X, en la cual cada fila corresponde a un término (i.e. una palabra lematizada) y cada columna a un documento (i.e. un subtítulo). El elemento x_{ij} de la matriz contiene la cantidad de veces que el término i aparece en el documento j (i.e. su frecuencia). Para poder obtener una medida de la importancia de cada término en nuestro corpus utilizamos el método TF-IDF (*Term Frequency - Inverse Document Frequency*) (Ramos et al., 2003), en el cual dada la colección de documentos D, una palabra w y un documento particular d ∈ D, se calcula:

$$\mathrm{w}_d = f_{\mathrm{w},d} \cdot \log\left(\frac{|D|}{f_{\mathrm{w},D}}\right)$$

Donde $f_{\mathrm{w},d}$ es la frecuencia de w en d (elemento original x_{ij} de la matriz X), $|D|$ corresponde al tamaño de la colección de documentos (número de columnas) y $f_{\mathrm{w},D}$ es la cantidad de documentos en los cuales aparece w (columnas diferentes de cero en la fila de w). Se puede ver que si una palabra aparece en todos los documentos entonces $|D| = f_{\mathrm{w},D}$: no es una palabra relevante para un documento particular y entonces la medida da 0. Por lo tanto, en cuantos más documentos aparezca la misma palabra menos importante resulta y por ende su medida se achica.

El próximo paso fue agrupar los documentos por la similitud semántica en tópicos, donde un tópico se define como grupo de documentos que, coloquialmente, "tocan el mismo tema". Más formalmente, un tópico se define como un conjunto de documentos cuyos vectores de frecuencias de palabras son (de acuerdo a alguna métrica adecuada)

similares; es decir, que comparten un contenido semántico similar. El método que utilizamos para determinar tópicos se basa en obtener la factorización no negativa de matrices de la matriz X de términos por documentos, luego de haber aplicado la transformación TF-IDF a los correspondientes elementos de matriz.

2.1.3. Factorización no negativa de matrices (NMF)

La factorización no negativa de matrices o NMF (por sus siglas en inglés[3]) es un método no supervisado para la detección de tópicos basado en la factorización de una matriz V en dos matrices W y H con la propiedad de que las matrices resultantes tienen elementos no negativos ($V \approx WH$) (Lee and Seung, 2001). Esta no-negatividad hace que las matrices sean más fácil de inspeccionar y útiles para la detección de tópicos, proveyendo una representación de los documentos en el espacio de tópicos (matriz W) donde cada columna indica el grado de pertenencia de cada documento a un dado tópico. Además, la matriz H provee la combinación de términos que describen cada tópico. El espacio de tópicos es usualmente mucho menor que el vocabulario (es decir, todos los términos que aparecen en el corpus de documento) y por lo tanto la matriz de "términos por documentos" se puede escribir como el producto de dos matrices de baja dimensionalidad: H de "tópicos por términos" y W de "documentos por tópicos".

Para obtener la factorización, se calculan W y H optimizando sobre una función objetivo, actualizando ambas matrices hasta la convergencia del algoritmo (Lee and Seung, 2001). La función objetivo para NMF se puede escribir como:

$$\frac{1}{2} \parallel V - WH \parallel_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} (V_{ij} - (WH)_{ij})^2$$

con $\parallel A \parallel_F^2 = \sum_{i,j} A_{ij}^2$ la norma Frobenius.

En esta función objetivo medimos el error de reconstrucción entre M y el producto de sus factores W y H basado en la distancia euclídea. Usando las reglas de actualización para W y H se pueden derivar y se obtiene:

$$W_{ic} \longleftarrow W_{ic} \frac{(VH^T)_{ic}}{(WHH^T)_{ic}} \quad ; \quad H_{cj} \longleftarrow H_{cj} \frac{(W^TV)_{cj}}{(W^TWH)_{cj}}$$

Los valores actualizados son calculados al mismo tiempo y usando las nuevas matrices W y H recalculamos el error de reconstrucción, repitiendo este proceso hasta la convergencia.

[3] Non-Negative Matrix Factorization

2.1.4. Ejemplo de modelado de tópicos con NMF

En esta sección vamos a mostrar cómo funciona el modelo NMF mediante un pequeño y sencillo ejemplo. Para ello, seleccionamos fragmentos oportunos (es decir, asegurando una coincidencia parcial del vocabulario) de los subtítulos de seis películas que se listan en la tabla 2.1, donde para completar la columna *GÉNERO* elegimos los primeros dos géneros que aparecen en sus respectivas páginas de Wikipedia (https://es.wikipedia.org/). Aclaramos que el resultado podría ser diferente si el modelo es aplicado a otros extractos de estos mismos subtítulos; también aclaramos que en el capítulo 5 usamos los subtítulos completos de las películas (no los fragmentos).

NOMBRE	AÑO	GÉNERO
1917	2019	Guerra, Épica
Amelie	2001	Comedia, Romance
Apocalypse Now	1979	Drama, Guerra
Avatar	2009	Ciencia Ficción, Aventura
Interstellar	2014	Ciencia Ficción
Titanic	1997	Romance, Drama

Tabla 2.1: Nombre, año y género de las películas utilizadas para el ejemplo.

42 00:04:41,864 → 00:04:44,741 It'd be a fresh start on a new world.	it had be a fresh start on a new world ['fresh', 'start', 'new', 'world']
43 00:04:45,034 → 00:04:48,704 You could do something important. You can make a difference.	you could do something important you can make a difference ['could', 'something', 'important'] ['make', 'difference']
44 00:04:53,960 → 00:04:55,877 And the pay is good.	and the pay is good ['pay', 'good']
45 00:04:56,504 → 00:04:57,504 Very good.	very good ['good']
46 00:04:58,756 → 00:05:00,632 Tommy was the scientist, not me.	tommy was the scientist not me ['tommy', 'scientist']
47 00:05:01,259 → 00:05:03,719 He was the one who wanted to get shot light-years out in space	he was the one who wanted to get shot light years out in space ['one', 'want', 'get', 'shot'] ['light', 'year', 'space']

Tabla 2.2: Fragmento de los subtítulos de la película "Avatar". En la primer columna se muestra el texto tal como se encuentra representado en los archivos *.srt*, y en la segunda parte se muestra el pre-procesamiento del texto y los tokens finales correspondientes a cada línea.

En la tabla 2.2 se muestra en la primera columna un fragmento de los subtítulos de la película **Avatar**, tal como aparecen en los archivos *.srt*. El primer número indica el número de línea del subtítulo, luego hay un intervalo temporal en el que aparece la línea en pantalla y luego aparece el texto de la misma. En la segunda columna se muestra el pre-procesamiento del texto. En la parte superior se expandieron las contracciones típicas del idioma, se cambiaron mayúsculas por minúsculas y se removieron signos de puntuación y otros símbolos; mientras que en la parte inferior aparece la línea ya pre-procesada. Es decir, se tokenizó la línea en cada palabra individual (*tokens*), se le asignó un *POS tag* a cada una de acuerdo a su rol gramatical en la oración, y luego con el token y el POS tag se lematizó la palabra llegando finalmente a su *lema* o *raíz*. En este último paso se omitieron los tokens que no correspondían a palabras del diccionario [4] o que eran *stop words*[5]. Si usted es un buen observador (y sabe inglés) se dará cuenta que la expansión de la primer línea de subtítulo que aparece en la tabla (la línea número 42) es incorrecta, ya que "**It'd** be a fresh start on a new world." debería expandirse a "**It would** be a ...". Esto se debe a que hay contracciones del idioma inglés que son ambiguas y pueden tener más de un significado. Elegimos la solución más directa, que consiste en reemplazar la contracción por la expansión más usual en el idioma inglés.

Una vez finalizado este procedimiento, obtuvimos una lista por película conteniendo todos los tokens y obtuvimos la representación TF-IDF aplicando la función **TfidfVectorizer** de la librería *scikit-learn* de Python (`https://scikit-learn.org/`), pidiendo que se retengan entre el 1 % y 99 % de las palabras menos y más usadas, respectivamente. Este procedimiento resulta en la matriz X de "términos por documentos", donde los términos son los tokens que aparecen en las seis películas y cada documento representa el texto de una película. Luego con la función **NMF** de la misma librería se calcularon las matrices H de "tópicos por términos" (dimensión 3×216) y W de "documentos por tópicos" (6×3) asignando el número de tópicos a tres ya que sabíamos de antemano que se podían distinguir estas películas con los tópicos 'Romance', 'Guerra' y 'Ciencia Ficción'.

Película	**Tópico 1**	**Tópico 2**	**Tópico 3**
Avatar	0	0	0.0278013
Apocalypse Now	1.12997	0	0
Titanic	0	1.28727	0
Interstellar	0	0	1.38997
1917	0.806441	0	0
Amelie	0	0.136173	0

TABLA 2.3: matriz W generada por el modelo NMF pasándole a la función como parámetro tres en número de tópicos.

[4] Diccionario inglés de la librería NLTK
[5] Es decir, que eran palabras de la lista "stop words" de la librería NLTK

Las fila i de la matriz W contiene información sobre el grado de pertenencia de la película i al tópico j. Este ejemplo tiene la particularidad que cada película tiene asignado un sólo tópico, aunque esto generalmente no es así. Como se puede ver en la tabla 2.3 **Apocalypse Now** y **1917** pertenecen al "tópico 1", **Titanic** y **Amelie** pertenecen al "tópico 2", **Avatar** e **Interstellar** pertenecen al "tópico 3". Dado nuestro conocimiento previo de los géneros de las películas (ver tabla 2.1) podemos llamar 'Guerra' al tópico 1 , 'Romance' al 2 y 'Ciencia Ficción' al último tópico.

La matriz H es más fácil de entender mediante *word clouds*, que consisten de imágenes con las palabras más frecuentes de cada tópico (Barth et al., 2014). A continuación, mostramos los *word clouds* de los tres tópicos y en ellos el tamaño de las palabras indican la importancia de éstas en el tópico, es decir, cuánto mayor tamaño tienen mayor es el grado de pertenencia al mismo.

FIGURA 2.1: Word cloud correspondiente al tópico 1 = 'Guerra'.

FIGURA 2.2: Word cloud correspondiente al tópico 2 = 'Romance'.

FIGURA 2.3: Word cloud correspondiente al tópico 3 = 'Ciencia Ficción'.

2.1.5. Evolución temporal del peso de los tópicos

Después de aplicar el algoritmo NMF, podemos ver cómo evoluciona el peso de los tópicos a lo largo del tiempo. El peso del tópico i (W_i) se define como la combinación de la cantidad de nuevas películas del tópico pesado por su grado de pertenencia. Puede ser definido año a año (resultando en una distribución dependiente del tiempo) o durante un período extendido en el tiempo (distribución promedio). Para un único año d, se muestra en la ecuación 2.1 la pertenencia por tópicos, donde w_{ji} (elemento ji de la matriz W) es el grado de pertenencia del documento j al tópico i; d_j es el año del documento j; y δ es la delta de Kronecker. Dado que cada documento puede tener todas las componentes no nulas, está permitido que un documento contribuya a más de un peso de los tópicos. Por último, normalizamos los perfiles temporales (Pinto et al., 2019).

$$W_i(d) = \sum_j w_{ji} \cdot \delta_{d_j,d} \tag{2.1}$$

2.1.6. Bump charts

Una manera de visualizar la evolución temporal del peso de los tópicos es a través de bump charts (Evergreen, 2019).

Un bump chart es una forma especial de gráfico de línea. Este tipo de gráficos está diseñado para explorar cambios en el ranking a través del tiempo. El enfoque aquí generalmente está en comparar la posición o el rendimiento de múltiples observaciones entre sí en lugar de los valores reales en sí.

Dicha visualización se puede usar para rastrear desde la tabla de clasificación en un torneo de fútbol, hasta la cantidad de búsquedas en Google de términos específicos relacionados a ciertos tópicos.

2.1.7. Visualización t-SNE

La Incrustación Estocástica de Vecinos (SNE por sus siglas en inglés[6]) es una herramienta para la visualización de datos de alta dimensionalidad (Maaten and Hinton, 2008). Dado un set de datos de alta dimensión $X = [x_1, x_2, ..., x_n]$ queremos mapearlo a dos o tres dimensiones de modo tal que $Y = [y_1, y_2, ..., y_n]$ pueda ser visualizado en un gráfico 2D. Nos referimos a la representación de baja dimensionalidad de la data Y como un mapa, y a la representación de baja dimensionalidad y_i de un dato como punto de mapa o *embedding*.

La similitud entre el dato x_j y el x_i es la probabilidad condicional, $p_{j|i}$, de que x_i elija a x_j como un vecino, si los vecinos fuesen elegidos en proporción a su densidad de probabilidad bajo una Gaussiana centrada en x_i.

$$p_{j|i} = \frac{exp(- \parallel x_i - x_j \parallel^2 /2\sigma_i^2)}{\sum_{k \neq i} exp(- \parallel x_i - x_k \parallel^2 /2\sigma_i^2)}$$

donde σ_i es la varianza de la gaussiana[7] centrada en x_i. Como solo nos interesa modelar la similitud entre dos datos seteamos $p_{i|i} = 0$. Análogamente para y_i e y_j es posible computar una probabilidad condicional similar que llamamos $q_{j|i}$. Elegimos la varianza de la Gaussiana igual a $1/\sqrt{2}$ y seteamos $q_{i|i} = 0$ como antes.

$$q_{j|i} = \frac{exp(- \parallel y_i - y_j \parallel^2)}{\sum_{k \neq i} exp(- \parallel y_i - y_k \parallel^2)}$$

Si la similitud entre los *embeddings* y_i e y_j modela bien la similitud entre los datos de alta dimensionalidad x_i y x_j entonces $p_{j|i} = q_{j|i}$. En base a esto el modelo SNE apunta a encontrar una representación que minimice la diferencia entre $p_{j|i}$ y $q_{j|i}$. Esta herramienta intenta minimizar la suma de las divergencias de Kullback-Leibler de todos los datos usando un método de descenso por el gradiente. La función de costo está dada por C.

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$$

[6] Stochastic Neighbor Embedding

[7] Para saber cómo determinar la gaussiana ver referencia (Maaten and Hinton, 2008)

Donde P_i representa la distribución de probabilidad condicional sobre todos los otros datos dado el dato x_i, y Q_i lo mismo para los *embeddings*.

Una alternativa a SNE es t-SNE[8] que minimiza una sola divergencia de Kullback-Leibler entre una distribución de probabilidad conjunta, P, en el espacio de alta dimensionalidad y una distribución probabilidad conjunta, Q, en el espacio de baja dimensionalidad.

$$C = KL(P_i||Q_i) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

con $p_{ii} = q_{ii} = 0$, $p_{ij} = p_{ji}$ y $q_{ij} = q_{ji}$ $\forall i, j$. En t-SNE la similitud entre un par de *embeddings*, q_{ij} es:

$$q_{ij} = \frac{exp(- \parallel y_i - y_j \parallel^2)}{\sum_{k \neq l} exp(- \parallel y_k - y_l \parallel^2)}$$

Y la manera obvia de definir la similitud entre dos datos x_i y x_j de alta dimensionalidad:

$$p_{ij} = \frac{exp(- \parallel x_i - x_j \parallel^2 / 2\sigma^2)}{\sum_{k \neq l} exp(- \parallel x_k - x_l \parallel^2 / 2\sigma^2)}$$

[8] distribución-t de Incrustación Estocástica de Vecinos

Capítulo 3

Métodos de representación de redes

"... behind each complex system there is an intricate network that encodes the interactions between the system's components."

Network Science by A. L. Barabási

Este capítulo está basado en el libro online "Network Science" de Albert-László Barabási (http://networksciencebook.com/), del cual proviene esta cita (Barabási et al., 2016).

3.1. Teoría de grafos

Nota sobre terminología: *en la literatura científica los términos* {red, nodo, enlace} *y* {grafo, vértice, arista} *pueden usarse indiferentemente, y así lo haremos en esta tesis. Sin embargo, mencionamos una pequeña diferencia en relación a estas dos terminologías: decimos redes cuando hablamos de sistemas reales, y grafos cuando discutimos la representación matemática o formal de estas redes.*

Si queremos entender a un sistema complejo, primero necesitamos conocer cómo sus componentes interactúan entre sí. Una forma de representar este tipo de sistemas es mediante la *teoría de grafos*, donde un grafo es una estructura matemática que está compuesta de *nodos* o *vértices* (las componentes del grafo), llamados **V**; y sus *enlaces* (interacciones entre componentes) **E**, que pueden ser tanto *dirigidos*, e.g. en una red de correos electrónicos una dirección de correo le escribe a otra, como *no dirigidos*, e.g. en

la red social Facebook dos usuarios que son 'amigos'. Los parámetros más básicos para describir los grafos son el número de nodos, **N**, y el número de enlaces, **L**. Si un grafo tiene todos sus enlaces dirigidos/no dirigidos, entonces decimos que se trata de un grafo dirigido/no dirigido.

Todos los grafos se pueden describir mediante su matriz de adyacencia **A** cuyo elemento a_{ij} representa la conexión entre el nodo *j=1,2,...,N* y el *i=1,2,...,N*. Es decir a_{12} vale 1 si hay un enlace dirigido del segundo nodo al primero y 0 si no. La matriz A de un grafo no dirigido es simétrica ($a_{ij} = a_{ji}$).

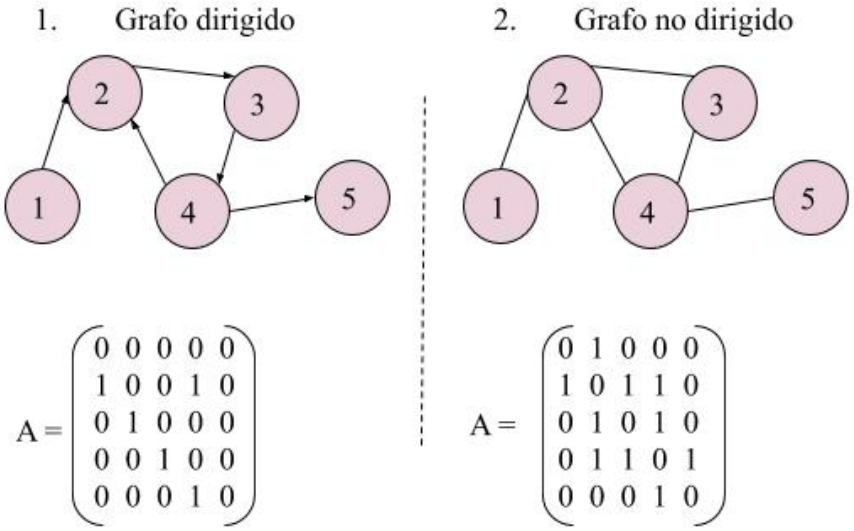


FIGURA 3.1: Esquemas de grafos dirigidos y no dirigidos con sus correspondientes matrices de adyacencia A.

Los gráficos que mostramos en la figura 3.1 se llaman *grafos binarios* pues los elementos de la matriz de adyacencia son 0 o 1. En cambio, si permitimos que $a_{ij} = w_{ij}$ con $w_{ij} \in \mathbb{R}$ pasan a ser *grafos pesados*, e.g. los pesos w_{ij} pueden ser los minutos que hablan dos personas (dos nodos) entre sí por teléfono. Muchas de las redes de interés científico son redes pesadas.

En grafos de sistemas reales **N** y **L** pueden variar ampliamente. Por ejemplo, la red neuronal del gusano *C. elegans* (el único sistema nervioso completamente mapeado de un organismo vivo), tiene $N = 302$ neuronas (nodos). Para contrastar, el cerebro humano se estima que tiene del orden de cien billones de neuronas $N = 10^{11}$ (Sporns et al., 2005). En una red de N nodos, la cantidad de enlances que puede tener el grafo es desde $L = 0$ hasta $L_{max} = \frac{N(N-1)}{2}$ cuando se trata de un grafo completamente conectado, es decir en cada nodo existen enlaces hacia todos los nodos.

3.2. Grafo generado a partir de matriz semántica

A partir de la matriz de "palabras por películas" en la representación TF-IDF generada por la función **TfidfVectorizer** de la biblioteca *sci-kit learn* en Python se puede construir la matriz W de "películas por tópicos" mediante el método NMF. Esta matriz W contiene la información del contenido semántico de las películas. Para cada película (vector fila de la matriz) tenemos los pesos asociados a los distintos tópicos. Por ejemplo si para el vector i, la componente j es nula significa que su contenido semántico no está representado por el tópico j. El grado de representación es directamente proporcional a la componente del vector.

Con la matriz W del contenido semántico del texto de los subtítulos de películas podemos computar su matriz de coeficientes de correlación de Pearson. La correlación de Pearson es una medida de la correlación lineal entre dos muestras de datos (nuestro caso las muestras son los vectores fila de la matriz W), su valor varía entre 1, una correlación perfectamente lineal y -1, correlación lineal negativa perfecta, pasando por 0 que indica que no hay correlación lineal entre las variables. Suponiendo que se están calculando dos variables aleatorias **X** e **Y**, el coeficiente de correlación de Pearson se simboliza con la letra $\rho_{x,y}$ y la expresión que nos permite calcularlo se muestra en la ecuación 3.1.

$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \tag{3.1}$$

donde σ_{XY} es la covarianza en (X,Y), σ_X es la desviación estándar de la variable X y σ_Y, la de la variable Y. De forma análoga, la ecuación se puede escribir en términos de μ_X, el promedio de la variable X, μ_Y, el de Y y E el valor de expectación. El coeficiente de correlación de Pearson aplicado a una muestra se suele notar $r_{x,y}$. Dados n datos emparejados $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, $r_{x,y}$ se define:

$$r_{x,y} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{3.2}$$

donde n es el número de muestras (o la dimensión de los vectores en nuestro caso), x_i son las muestras individuales y $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ es el promedio de x que se define análogamente para el set de muestras y_i.

Volviendo a la construcción del grafo vemos que $r_{x,y}$ nos da una medida de la correlación lineal entre los vectores $x = [x_1, x_2, ..., x_n]$ e $y = [y_1, y_2, ..., y_n]$. Al calcular la correlación de Pearson de la matriz W obtenemos entonces la matriz A de correlación

del contenido semántico cuyos elementos son los $r_{i,j}$ para $i = j = 1, ..., N$ con N=54.731 el número de nodos. Por como está definida la correlación de Pearson, A es una matriz simétrica ya que $r_{i,j} = r_{j,i}$. Esto hace que la red de asociación semántica sea una red no dirigida (si hay un enlace entre las películas 1 y 4, hay otro entre las películas 4 y 1) y además puede considerarse que el grafo está completamente conectado.

3.3. Distancia y caminos en grafos

La noción de distancia es fundamental en las ciencias físicas para estudiar un determinado sistema. Por ejemplo, para conocer la fuerza gravitatoria que sienten dos cuerpos celestes o la fuerza electromagnética entre dos partículas cargadas es necesario saber la distancia espacial entre los dos objetos. Sin embargo, cuando trabajamos con grafos una medida de distancia debe ser inferida en este espacio. ¿Qué es la distancia entre dos personas que no se conocen, o entre dos palabras?, en estos casos no tiene sentido pensar en la *distancia espacial* pues dos personas podrían estar literalmente al lado una de la otra y no conocerse. Dos palabras podrían estar escritas de forma consecutiva en una oración o ser las más usadas en un texto y ser sinónimos, antónimos, de distinto tipo, etc.

Por ello cuando hablamos de redes inferimos una medida de diferencia a partir de interconectividades entre nodos llamada *longitud de camino*. Un *camino* es una manera de ir de un nodo de la red a otro a través de enlaces, y su longitud es la cantidad de enlaces por los que pasa el camino. Algunas veces se pide que el camino pase una sola vez por cada nodo visitado.

3.3.1. Camino mínimo

El camino mínimo entre los nodos i y j se define como el camino que une ambos nodos y posee la menor cantidad de enlaces - a veces se lo llama también la distancia entre esos dos nodos, d_{ij}. Puede haber muchos caminos mínimos de la misma longitud d_{ij} entre un par de nodos, pero ninguno de ellos contiene loops (i.e. ir y volver al mismo nodo, que puede pasar si en la matriz de adyacencia hay un 1 en la diagonal) o intersectarse a sí mismo (i.e. pasar por algún nodo dos veces).

En una red no dirigida $d_{ij} = d_{ji}$, es decir, que la distancia entre el nodo i y el nodo j es la misma que entre el nodo j y el i. En una red dirigida puede pasar que $d_{ij} \neq d_{ji}$ y hasta podría llegar a no existir un camino entre el nodo j y el i.

En un grafo no dirigido y completamente conectado, hay del orden de $(N-2)!$ caminos entre un par determinado de nodos, donde N es la cantidad de nodos en el grafo. Si el número de nodos es grande, listar exhaustivamente todos los caminos se vuelve prohibitivo computacionalmente. Existen algoritmos para determinar el camino más corto existente entre dos nodos de una red, tales como el algoritmo de Dijkstra, el cual requiere en su implementación original una cantidad de pasos del orden de N^2. Dado el tamaño considerable de la red de asociación semántica entre subtítulos, exploramos un método diferente para evaluar si un determinado camino (por ejemplo, el cronológico) se aproxima al camino más corto posible entre dos nodos determinados.

3.3.2. *Ensemble* de caminos en grafos

En esta sección vamos a explicar el método que utilizamos para evaluar la distancia de caminos en el grafo generado por la matriz de adyacencia obtenida a partir de la correlación de Pearson entre las filas de la matriz W (ver sección 3.2).

Como ya dijimos el grafo que construimos para la red de asociación semántica es no dirigido, consta de N nodos y es completamente conectado. Cada nodo representa el contenido semántico de los subtítulos de una película y tiene dos propiedades (o metadatos): la *fecha* (el año de estreno de la película); y el tópico que le asignamos por medio de la detección de tópicos utilizando el modelo NMF (veremos esto detalladamente más adelante). Para estudiar la relación que existe entre el orden cronológico (ordenamiento por *fecha*) de las películas en la longitud de caminos del grafo, medimos la longitud del camino entre el nodo más antiguo y el más contemporáneo pasando por todos los nodos del grafo una única vez en **orden cronológico**. A continuación realizamos una permutación aleatoria[1] de los nodos, conservando el punto de partida, el nodo más antiguo, y el de llegada, el nodo más contemporáneo. Repetimos este último procedimiento miles de veces creando un "ensemble" de caminos en el grafo y comparamos sus longitudes con la del camino cronológico. Este método puede ser adaptado para entender como otras propiedades del grafo dependen del ordenamiento temporal, así como modificado para forzar nodos intermedios que los caminos en el ensemble deben atravesar (además del primer y último nodo, en el sentido cronológico).

En la figura 3.2 se muestra un esquema del procedimiento detallado anteriormente. Para estudiar la segunda propiedad de los nodos comparamos el camino entre un nodo con determinado tópico y otro nodo con distinto tópico pasando por todos los nodos del mismo tópico primero y luego saltando a otro tópico.

[1] Al realizar una permutación en el orden de los nodos, el grafo ya no es recorrido de forma cronológica

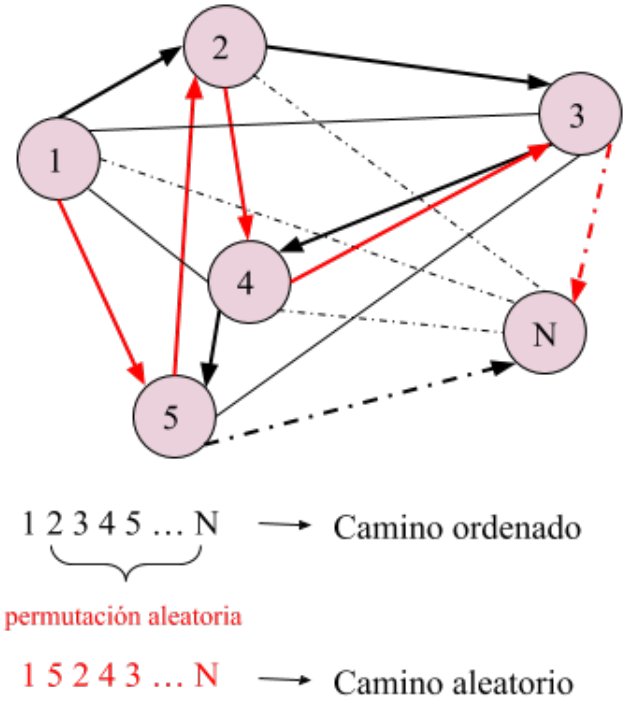


FIGURA 3.2: Camino ordenado y aleatorio de un grafo. Las etiquetas 1, 2, ..., N corresponden a un determinado ordenamiento de los nodos, que puede ser tanto cronológico como por tópicos.

Capítulo 4

Descripción de los datos y estadística

En este capítulo contaremos la procedencia de la base de datos de subtítulos de películas que utilizamos. Luego, describiremos su composición en la sección 4.1 y mostraremos, indicando en los histogramas correspondientes, los filtros usados para quedarnos sólo con los subtítulos de películas deseados para el análisis siguiente.

ACLARACIÓN: *llamaremos 'película' al archivo correspondiente a los subtítulos de una película o capítulo de serie de la base de datos de opensubtitles.*

4.1. Descripción de la base de datos

Como ya mencionamos anteriormente, la base de datos que estudiamos en esta tesis proviene de la página `http://www.opensubtitles.org/` que accedió a proveernos los archivos pertenecientes a su página web por una pequeña donación y nuestro compromiso de citar la fuente de los datos. Luego de recibir una módica suma de dinero compartieron 12Gb de datos, conteniendo 570.000 archivos *IDdelSubtítulo.srt*. También se incluía un archivo *.txt* con la metadata de la base de datos: el ID de la película; el ID del subtítulo; el nombre de la película; el año de publicación; el ID y nombre del usuario que subió el mismo a la página, etc.

Creamos un iterador en Python en el cual cada ciclo tenía como input el texto de una fila escrito en el *.txt* con la metadata y que poseía los datos de un único archivo. Confeccionamos una lista con el nombre de película, año de publicación, ID del subtítulo de todos los archivos siempre que hubiese un cambio en el ID de la película y que el

año de publicación fuese mayor a 1895[1]. Esta última condición la pusimos para evitar trabajar con películas que no tenían asignado un año de publicación, que este no fuese correcto o que estuviese deliberadamente mal, por ejemplo "Justin Bieber's ultimate #& %$ collection" que tenía asignado como año de publicación 666. De esta manera redujimos la base de datos a 170.000 archivos con subtítulos pertenecientes a películas o capítulos de serie **distintos**.

Como dijimos en el capítulo 1 la ventaja de estudiar la semántica en subtítulos de películas es que cada línea está etiquetada temporalmente, lo que nos permite estudiar la evolución en el tiempo (a diferencia, por ejemplo, de un libro). Es por ello que queríamos obtener el texto escrito en cada línea del subtítulo y el intervalo temporal en el que aparecía este en pantalla. La primer complicación apareció en este proceso debido a la enorme cantidad de archivos y la dificultad de comprobar el contenido de todos ellos manualmente. Si bien todos poseen un formato similar al que se ve en la figura 4.1 en algunos subtítulos las fracciones de segundos en los intervalos temporales estaban separados por distintos caracteres (por ejemplo, '.' en vez de ','). En muchos otros los números de línea del subtítulo no eran enteros consecutivos, o algunas *líneas* (llamando líneas a lo que está entre el número de línea y el texto de la misma) separadas por ningún o más de un espacio en blanco. A estas particulares se les suma el hecho de que los subtítulos están escritos por humanos que pueden cometer errores de tipeo. La manera que encontramos más consistente para adquirir el texto e intervalo temporal de los archivos fue hacer un split[2] buscando la secuencia de caracteres '$--> $' y creando una función que toma como parámetros la hora escrita en formato HH:MM:SS y devuelve la cantidad de segundos correspondientes. Para el texto de las líneas consideramos todo el texto comprendido entre dos '$--> $' ya que parte del preprocesamiento del texto (ver sección 2.1.2) implica remover los caracteres numérico y por lo tanto eliminar el número de línea del subtítulo.

Utilizamos la biblioteca para el análisis de datos de código abierto **Pandas** (`https://pandas.pydata.org/`) (McKinney and Team, 2015). Construimos un *dataframe*, una clase de datos utilizada en Pandas que es similar a una tabla indexada pero con la ventaja que las entradas pueden ser heterogéneas, es decir pueden ser objetos tipo listas, constantes, vectores, strings, etc. Las columnas del dataframe contenían el ID del subtítulo, el nombre de la película, el año de publicación de la misma, una lista donde cada elemento es una línea del subtítulo ya pre-procesada correspondiente a ese archivo y otra lista cuyos elementos son intervalos temporales, tiempo inicial y tiempo final, en segundos, donde aparecía cada línea en pantalla. Notar que las últimas dos columnas

[1] Año en el cual comenzó la historia del cine como espectáculo (`https://es.wikipedia.org/wiki/Historia_del_cine`).

[2] El método **split('ejemplo')** de Python 3.1 'corta' un texto cuando encuentra la palabra 'ejemplo' en el mismo y junta todas las partes que hayan sido cortadas en una lista.

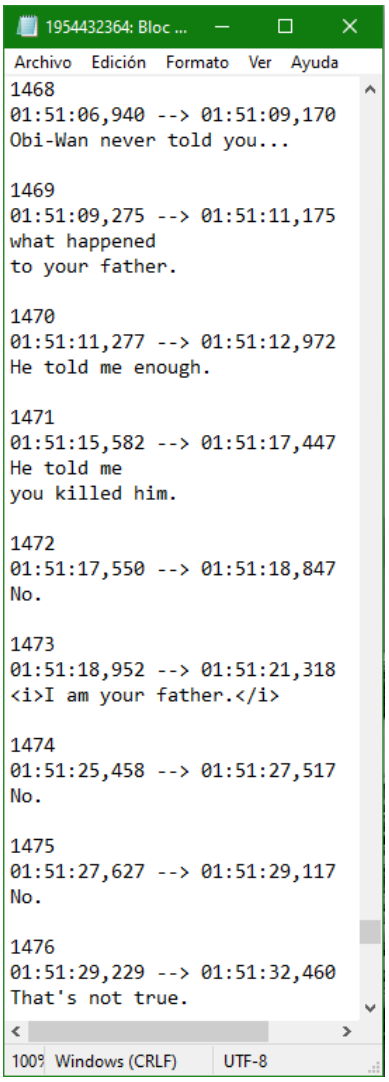


FIGURA 4.1: Fragmento del contenido de uno de los archivos de subtítulos *.srt*.

contienen una lista por cada entrada (película) del dataframe, es decir, que las columnas enteras son dos series de listas.

4.2. Histogramas

Para caracterizar nuestra base de datos con el texto contenido en los 170.000 archivos asociados a películas diferentes, realizamos una serie de histogramas que mostraremos a continuación. Basándonos en ellos creamos filtros para la duración de los subtítulos y para la cantidad de palabras que se encuentran en los subtítulos por película. Al aplicar estos filtros en la base de datos nos quedaron aproximadamente 50.000 películas, casi el 10 % de los archivos iniciales de la base de opensubtitles y un poco menos de un tercio de los archivos asociados a películas distintas.

El primer histograma que confeccionamos fue el de cantidad de películas por año, que se puede ver en la figura 4.2.

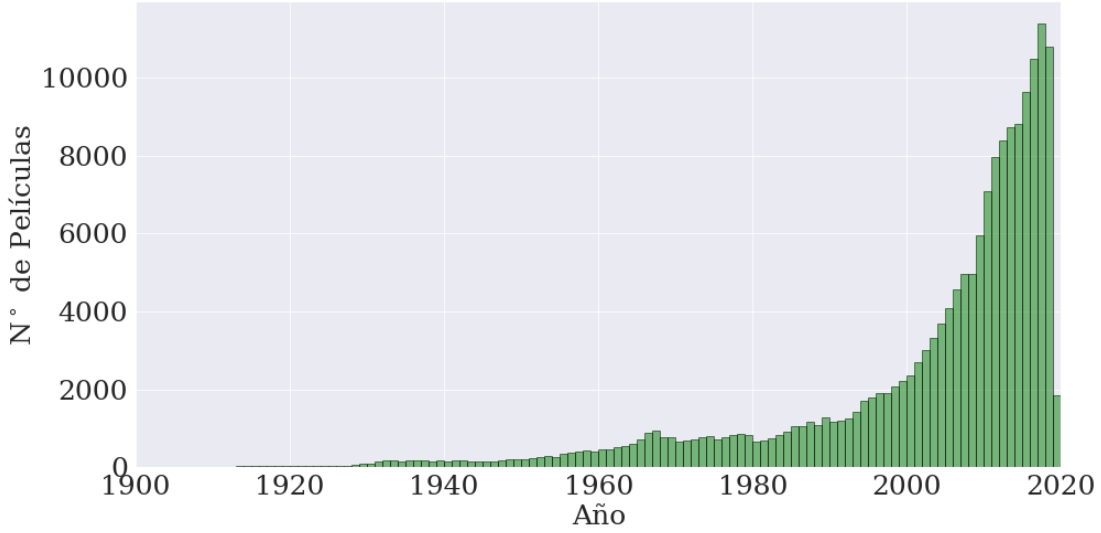


FIGURA 4.2: Histograma de frecuencia de cantidad de películas por año.

Como era de esperar la cantidad de películas crece enormemente en las últimas dos décadas. Dentro de este fenómeno se encuentra el auge de las series cuyos capítulos tienen menor costo de producción que los de una típica película hollywoodense.

4.2.1. Cantidad de palabras por película

Utilizando el dataframe construido con nuestra base de datos (ver la sección anterior 4.1), implementamos una función que dada la columna del dataframe con la lista de las líneas de subtítulos cuenta la cantidad de tokens en las películas. En la figura 4.3 mostramos el histograma de frecuencia de la cantidad de palabras por película en bines de a 100 palabras, es decir el primer bin comprende las películas que tienen entre 0 y 99 palabras, el segundo bin las que poseen entre 100 y 199, y así sucesivamente.

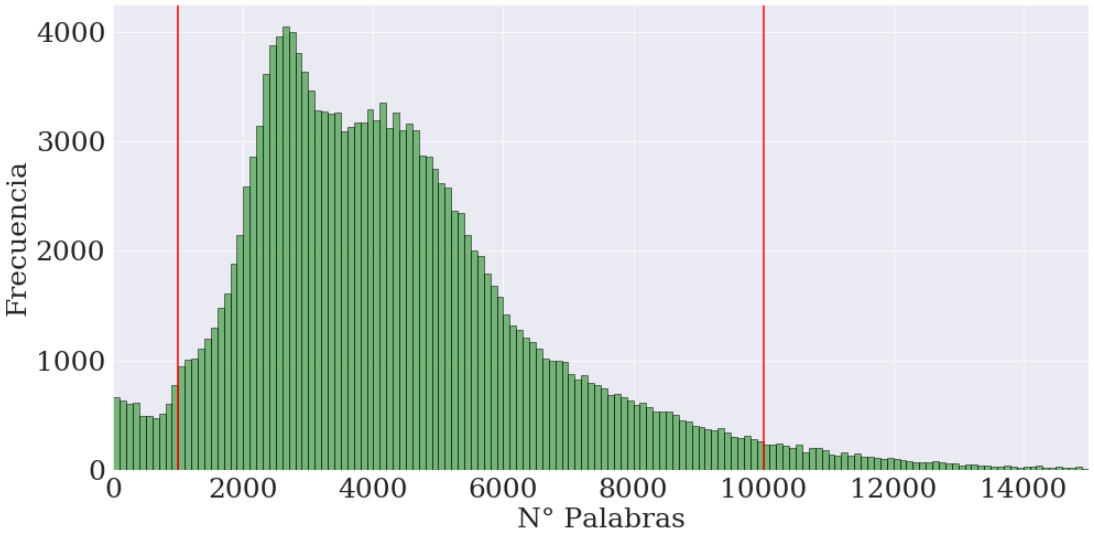


FIGURA 4.3: Histograma de frecuencia de cantidad de palabras en los subtítulos en bines de 100 palabras. Indicamos con líneas rojas el filtro inferior, de 1.000 palabras, y superior, de 10.000, para la cantidad de palabras por película.

Podemos observar en el histograma un mínimo local cerca de las 500 palabras seguido de una campana no gaussiana que termina con una cola larga que va tendiendo a 0. Para quedarnos con la mayor cantidad de películas posibles a la vez que eliminábamos los subtítulos que correspondían a películas mudas, con subtítulos escritos en otros idiomas, subtítulos con palabras que describen escenas en vez de diálogos, etc., pusimos límites superior e inferior, de 1.000 y 10.000 palabras respectivamente, para la cantidad de palabras por subtítulo. Verificamos la validez de estos límites inspeccionando manualmente varios casos de películas excluidas.

4.2.2. Duración de los subtítulos

Tomamos como duración de los subtítulos de una película el valor máximo de la columna del dataframe donde se encuentra la lista de los segundos donde aparecen las líneas de subtítulo en pantalla, e.g. si el valor máximo es 6.250 [segundos] entonces le asignamos a la película una duración de 104 [minutos]. En la figura 4.4 se puede observar el histograma de frecuencia que confeccionamos para visualizar la distribución, en bines de un minuto, de la duración de los subtítulos.

Se pueden observar picos alrededor de los minutos 22, 48 y 59 donde se encuentran la gran mayoría de subtítulos correspondientes a capítulos de series. También se ve que hay películas que duran muy pocos minutos donde se encontraban por ejemplo cinemáticas de videojuegos, subtítulos incompletos, etc. Como el objetivo de este trabajo es estudiar la evolución temporal del contenido semántico en películas creamos un filtro inferior de

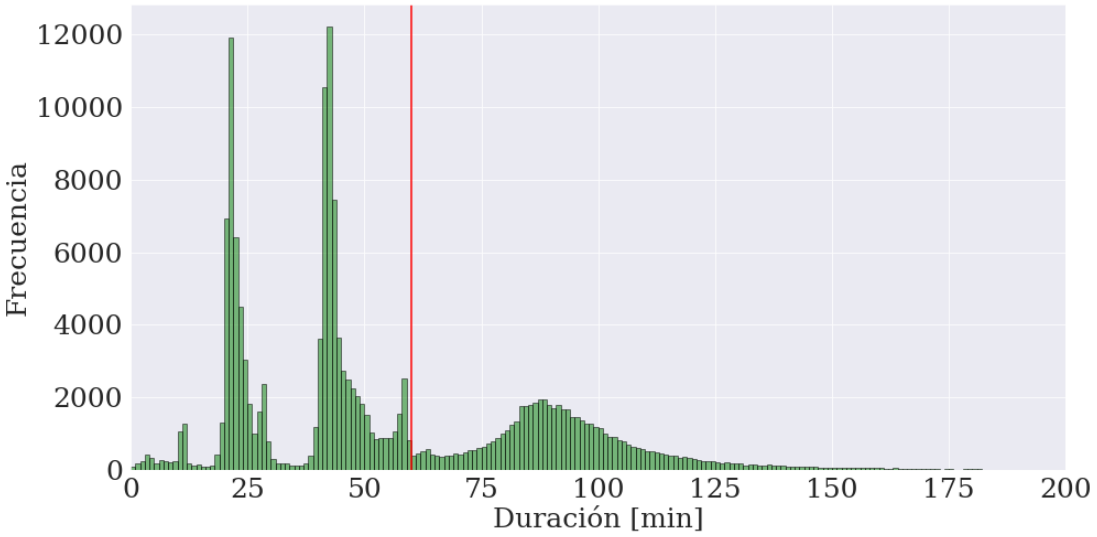


FIGURA 4.4: Histograma de frecuencia de la duración de los subtítulos en bines de 1 minuto. Para calcular la duración tomamos el valor máximo de la lista con los segundos donde aparecen las líneas de subtítulos en la pantalla. También indicamos con una línea roja el filtro inferior de 60 minutos.

60 minutos para quedarnos sólo con los subtítulos que duran al menos una hora. Algunos usuarios ponían su firma en el texto de los archivos al final de los subtítulos de la película y elegían un intervalo temporal muy superior a la duración real de la película, por lo que también pusimos un filtro superior de 300 minutos. La idea de este filtrado es quedarnos con la campana de películas con centro en aproximadamente 90 minutos.

4.3. Dataframe filtrado

Recordamos que cada entrada del dataframe contiene la siguiente información: nombre de la película, año de publicación, una lista cuyos elementos son listas de los tokens de cada línea de subtítulos, y otra lista particular. Esta última lista difiere en que en lugar de los tokens de las líneas, contiene la información del intervalo temporal en el cual aparece la línea en pantalla.

Con la información de los histogramas de la sección anterior creamos dos filtros para el dataframe correspondiente a los datos de los 170.000 archivos de películas distintas provenientes de opensubtitles. Eliminando las entradas de los subtítulos de películas que no tuvieran entre 1.000 y 10.000 palabras (tokens) y que duraran menos de 60 o más de 300 minutos. De esta forma redujimos el dataframe original al dataframe filtrado con los datos de 54.731 subtítulos de películas. De los cuales obtuvimos casi 55 millones de líneas de subtítulos que contenían más de 300 millones de tokens. La duración total de los subtítulos de películas del dataframe filtrado llega a un poco menos 85 mil minutos que son equivalentes a más de 3.500 días (o más de 9 años y medio).

Capítulo 5

Resultados de la detección de tópicos mediante NMF

En este capítulo mostraremos los resultados que obtuvimos al aplicar el método de factorización de matrices no negativa, NMF (ver capítulo 2 sección 2.1.3), al texto de los subtítulos de las películas de nuestra base de datos. Primero contaremos cómo elegimos el número de tópicos con el que el método separa la matriz de "palabras por películas" en las matrices de baja dimensionalidad "tópicos por palabras" y "películas por tópicos". Luego, incluiremos las *word clouds* e histogramas de los tópicos resultantes. También confeccionamos *bump charts* (ver capítulo 2 sección 2.1.6) para la evolución temporal del peso de los tópicos. Y, en último lugar, mostraremos una visualización en 2D realizada con la distribución-t de Incrustación Estocástica de Vecinos, t-SNE, de la evolución temporal de la cantidad de películas por tópicos.

Usando el dataframe filtrado con los datos de 54.731 *películas filtradas*[1] construimos una lista de strings[2] donde cada elemento de la lista contenía todos los tokens presentes en el subtítulo de una película. Esta lista es el input de la función **TfidfVectorizer** de la biblioteca sci-kit learn de Python (recordamos el capítulo 2, sección 2.1.4) que computa la matriz A con los coeficientes TF-IDF de la matriz "palabras por películas". Es decir dada la totalidad de tokens presentes en las películas (todas las palabras que aparecen en el corpus de texto de los subtítulos de todas las películas) crea un "vocabulario", una matriz de dimensión vocabulario por número de películas y calcula el coeficiente TF-IDF a cada elemento de la matriz. Además, la función permite ignorar los tokens más y menos frecuentes del vocabulario en el corpus de texto.

[1] ver capítulo anterior 4

[2] es un tipo de objeto en Python para manejar texto https://docs.python.org/3/library/string.html

Pidiendo que se retengan entre el 5 % y 95 % de las palabras menos y más usadas, respectivamente, obtuvimos la matriz A de "palabras por películas" en representación TF-IDF. Esta contiene los datos a entrenar por el método NMF que la descompone en dos matrices de menor dimensión "tópicos por palabras" (dimensión N×2.766) y "películas por tópicos" (dimensión 54.731×N) con N el número de tópicos siendo un parámetro de la función. Veremos en la siguiente sección cómo elegimos este número.

5.1. Elección del número de tópicos en el método NMF

Consideremos el problema de dividir un texto en un número de tópicos. El problema es comparable a otros relacionados con clusterización de datos, división de un grafo en módulos, etc., y requiere de un criterio para definir la cantidad de grupos en los cuales se dividirán los datos. Existen distintos criterios más o menos sofisticados para determinar este número, por ejemplo, criterios basados en teoría de la información o el método de la silueta. Dado que el número de tópicos no es una variable fundamental en nuestro análisis, decidimos basar el número de tópicos utilizando dos criterios en simultáneo: primero, la inspección visual de los tópicos, su coherencia y su redundancia como función de su cantidad; segundo, la determinación del momento en el cual cambia la pendiente del error de reconstrucción del método NMF como función de la cantidad total de tópicos retenidos (también conocido como criterio o método del codo o "elbow method"). Ambos criterios resultaron en un número final de 10 tópicos, que justificamos a continuación.

Utilizamos el *elbow method* para determinar el número de tópicos en los que clusterizar el contenido semántico de los subtítulos de las películas de nuestra base de datos (Thorndike, 1953).

Aprovechamos el atributo *reconstruction_err_* de la función **NMF()** de la librería SCIKIT LEARN de Python. Este error de reconstrucción es un número que indica la norma Frobenius de la matriz diferencia entre los datos de entrenamiento, i.e. una lista donde cada elemento contiene las palabras de una película de nuestra base de datos, y la data reconstruida con el producto de matrices WH del modelo ajustado.

Calculamos el error de reconstrucción de la función **NMF** proveniente de ajustar el modelo a nuestros datos de entrenamiento para distintos valores en el parámetro la función 'número de tópicos'. En un primer intento por encontrar un valor idóneo, confeccionamos un gráfico para distintos valores, entre 1 y 5.000, del parámetro. Para valores superiores a 500 tópicos la pendiente del error se volvía positiva (i. e. aumentaba) hasta llegar a un valor de saturación. En la figura 5.1 podemos notar que no hay un

cambio muy significativo de la pendiente del error de reconstrucción pero vemos un cambio sutil en los primeros 25 valores para 'N° Tópicos'.

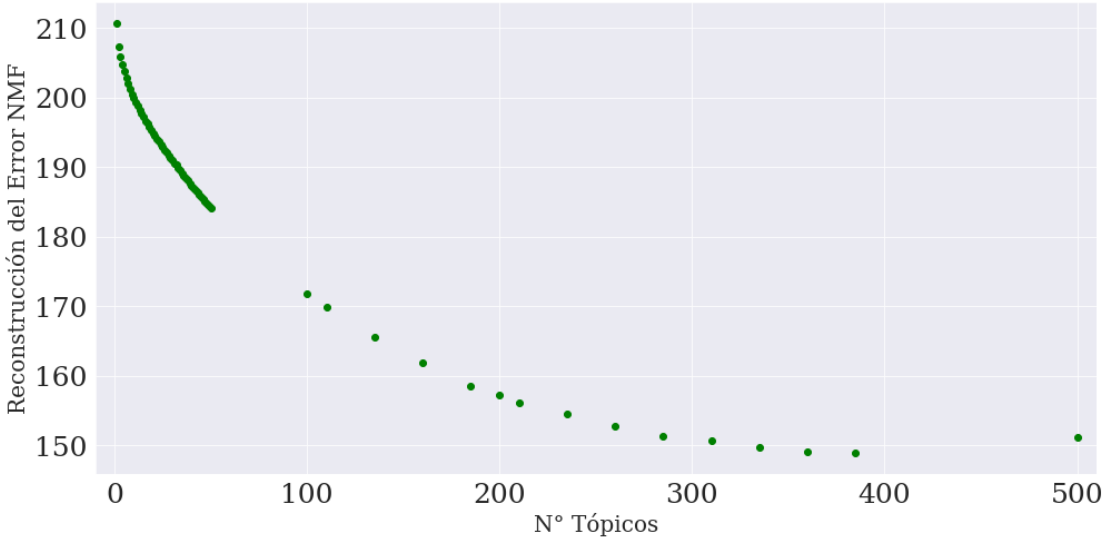


FIGURA 5.1: Gráfico del error de reconstrucción de la función NMF() de la librería sci-kit learn de Python aplicada a nuestra base de datos con el texto de los subtítulos de películas.

Por ello realizamos un barrido de número de tópicos más detallado entre los valores 1 y 50. Graficamos la derivada de la función error para estos valores mediante la función **gradient**[3] sobre los datos del error de reconstrucción que se puede ver en la figura 5.2.

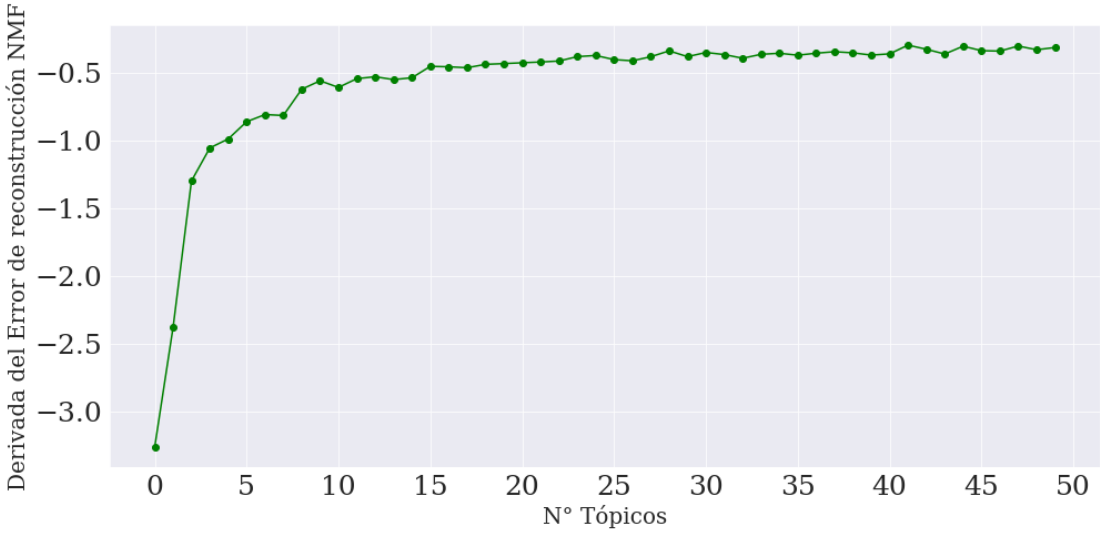


FIGURA 5.2: Gráfico de la derivada del error de reconstrucción de la función NMF() de la librería sci-kit learn de Python aplicada a nuestra base de datos con el texto de los subtítulos de películas.

[3]Esta función calcula el gradiente de un vector de datos mediante diferencias centrales precisas a segundo orden en los puntos interiores del vector y a primer o segundo orden en los bordes (https://github.com/numpy/numpy/blob/v1.17.0/numpy/lib/function_base.py#L801-L1140)

Notamos un codo entre los valores 7 y 10 de 'N° Tópicos' en la derivada de la función error. Para un mayor número de tópicos se puede apreciar que la derivada de la función error tiene un comportamiento lineal de pendiente menor a la de los puntos anteriores. La idea detrás del *elbow method* es que agregar tópicos disminuye considerablemente el error del método NMF, pero al ir aumentando el número de tópicos, el error se reduce en menor medida hasta el punto en que el costo de agregar un tópico más no justifica lo poco que se gana reduciendo el error.

Comparamos las *word clouds* generadas por el método para un mayor número de tópicos (13, 20 y más) y vimos que algunas eran redundantes. Entonces, por lo mencionado antes, el valor que elegimos para la división de tópicos con distinto contenido semántico en el texto de los subtítulos de películas fue 10. Es decir pedimos que el método NMF descomponga a la matriz de "palabras por película" (2.766×54.731) en representación TF-IDF en dos matrices de menor dimensión W y H. La matriz H, de dimensión "tópicos por palabras" (10×2.766), contiene la información del conjunto de palabras presente en cada uno de los 10 tópicos y su respectivo peso. A su vez, la matriz W, de dimensión "películas por tópicos" (54.731×10), posee el grado de pertenencia de las películas a los distintos tópicos. Al igual que en el ejemplo de la introducción, elegimos representar a la matriz H de forma visual a través de *word clouds*.

5.2. Nombre de los tópicos

En esta sección contaremos cómo elegimos los nombres de los 10 tópicos que obtuvimos aplicando el algoritmo NMF a la representación TF-IDF de los subtítulos.

Buscamos en la base de datos de películas online IMDb https://www.imdb.com/ los géneros y resúmenes que allí aparecen de las diez películas más representativas de cada tópico. Es decir, para el tópico j nos quedamos con los diez mayores valores de la columna j de la matriz W de "películas por tópicos", con el índice de fila i correspondiente a esas componentes ubicamos las diez películas más representativas por cada tópico.

A continuación incluiremos tablas 5.1 a 5.10 con el nombre, año y género (de IMDb) de las diez películas más representativas de cada tópico, indicando el nombre que le pusimos a cada uno. Es necesario aclarar que la idea fue tratar de obtener un clasificador de género en base al contenido semántico de los subtítulos de las películas. Por ese motivo llamamos a los tópicos en base a los géneros de las tablas y del resumen de la película. Sin embargo podría ser más apropiado que los tópicos se refieran por ejemplo a "boda", "realeza", "navidad" o "religión". Aunque para eso se necesitaría tener un conocimiento

previo como una metadata que contenga temáticas, porque si la quisiéramos construir nosotros manualmente deberíamos estar (literalmente) 9 años y medio de corrido viendo las películas de nuestra base de datos. Notamos que IMDb posee una API ("Application Programming Interface") que permite recuperar el género de una película dado su nombre; no obstante su utilización para asignar sistemáticamente género a todas las películas en la base de datos se encuentra por fuera del alcance de la tesis, dado que existen numerosos problemas técnicos que deben ser resueltos antes. El principal es que existen numerosas películas con el mismo nombre o variantes que pueden ser confundidas por la API (incluso para películas del mismo año). Además, la API de IMDb es gratis únicamente para menos de 1000 requests por día.

NOMBRE	AÑO	GÉNERO
Miami Rhapsody	1995	Comedia
Deux	2015	Drama, Historia, Romance
La seduzione	1973	Crimen, Drama, Romance
La femme defendue	1997	Crimen, Drama, Romance
Three Murderesses	1959	Comedia
Tutta colpa di Freud	2014	Comedia
Cristina Quer Casar	2003	Comedia, Romance
Questi giorni	2016	Drama
Va voir maman, papa travaille	1978	Comedia
42plus	2007	Romance, Drama

Tabla 5.1: Nombre, año y género de las películas más frecuentes en el tópico 0: Romance/Comedia.

NOMBRE	AÑO	GÉNERO
The Exploding Girl	2009	Drama
Fairhaven	2012	Comedia, Drama
Loves Her Gun	2013	Drama
Donald Cried	2016	Comedia, Drama
Kate Can't Swim	2017	Drama
The Sleepwalker	2014	Drama
Dean	2016	Comedia, Drama, Romance
Uncle John	2015	Crimen, Drama, Misterio
Newness	2017	Drama, Romance
Drinking Buddies	2013	Comedia, Drama, Romance

Tabla 5.2: Nombre, año y género de las películas más frecuentes en el tópico 1: Drama/Romance.

En algunos casos, el nombre elegido para los tópicos no tuvo mucho debate porque todas o la mayoría de las películas más representativas del mismo compartían uno o más géneros de acuerdo su página en IMDb. El más claro de todos fue la elección del nombre para el tópico 2 (ver tabla 5.3) en el cual todas las películas incluían el género

NOMBRE	AÑO	GÉNERO
Shadow World	2016	Documental
Hell on Earth: The Fall of Syria and the Rise of ISIS	2017	Documental
"The World Wars" A Rising Threat	2014	Documental, Historia, Guerra
Auserwahlt und ausgegrenzt - Der Hass auf Juden in Europa	2017	Documental
Krieg und Frieden	1982	Documental
"The Vietnam War" Riding the Tiger (1961-1963)	1961	Documental, Historia, Guerra
Comandante	2003	Documental
Ethos	2011	Documental
My Enemy's Enemy	2007	Documental, Biografía, Historia
Our Man in Tehran	2013	Documental, Historia

Tabla 5.3: Nombre, año y género de las películas más frecuentes en el tópico 2: Documental.

NOMBRE	AÑO	GÉNERO
Wallace and Gromit: The Best of Aardman Animation	1996	Comedia, Animación
Day-Time Wife	1939	Comedia, Romance
Popeye	1980	Aventura, Comedia, Familia
Feet First	1930	Comedia
Back When We Were Grownups	2004	Drama
Sylvia Scarlett	1935	Comedia, Romance, Drama
Carry on Behind	1975	Comedia
No Limit	1931	Comedia, Drama, Musical
Daddy Long Legs	1955	Musical, Romance
Move Over, Darling	1963	Comedia, Romance

Tabla 5.4: Nombre, año y género de las películas más frecuentes en el tópico 3: Comedia.

NOMBRE	AÑO	GÉNERO
Patterns	1956	Drama
Pride and Prejudice	2005	Drama, Romance
Mystery of Edwin Drood	1935	Drama, Horror, Misterio
Devotion	1931	Comedia, Drama, Romance
In the Mood for Love	2000	Drama, Romance
"Mr Selfridge" Episode 1.1	2013	Drama
"All Creatures Great and Small" 1985 Special	1985	Comedia, Drama
Singapore	1947	Acción, Aventura, Crimen
The Woman in Black	1989	Horror, Misterio
High and Dry	1954	Comedia

Tabla 5.5: Nombre, año y género de las películas más frecuentes en el tópico 4: Drama.

'Documental'. En cambio los tópicos 0[4], 1, 3 y 4 aparecen mucho los géneros 'Comedia', 'Drama' y 'Romance', para esos casos tratamos de darle prioridad a los géneros de la primer película más representativa del tópico, o nos ayudamos con el resumen escrito de

[4]El tópico 0 es el primer tópico. Tiene esta notación porque el primer elemento de una lista o vector en Python tiene índice 0.

NOMBRE	AÑO	GÉNERO
Kenju zankoku monogatari	1964	Acción, Crimen
Youth of the Beast	1963	Acción, Crimen, Misterio
X	2002	Acción, Aventura, Thriller
Mi cheng	2015	Acción, Crimen, Thriller
Nommer 37	2018	Thriller
The Girl from the Naked Eye	2012	Acción, Crimen, Misterio
Lucio Flavio, o Passageiro da Agonia	1977	Crimen, Drama
Shikingen godatsu	1975	Crimen
Gen-X Cops	1999	Acción, Comedia, Crimen
Talento de barrio	2008	Acción, Crimen, Drama

TABLA 5.6: Nombre, año y género de las películas más frecuentes en el tópico 5: Acción/ Crimen.

NOMBRE	AÑO	GÉNERO
Quarries	2016	Crimen, Drama, Thriller
Static	2012	Drama, Horror, Misterio
Or	2004	Drama, Romance
Lovely Loneliness	2008	Comedia, Romance
Splinter	2008	Horror, Ciencia Ficción, Thriller
Girlfriend	2010	Drama
Zombie Night	2013	Horror, Thriller
Blue Valentine	2010	Romance, Drama
Scalene	2011	Crimen, Drama, Thriller
The Royal Variety Performance 2000	2000	Comedia, Musical

TABLA 5.7: Nombre, año y género de las películas más frecuentes en el tópico 6: Horror/ Thriller.

NOMBRE	AÑO	GÉNERO
Yongseobadji mothan ja	2005	Drama
Torpedo Run	1958	Drama, Guerra
404	2011	Drama, Misterio, Thriller
U-Turn	2018	Misterio, Thriller
Horatio Hornblower: The Fire Ship	1998	Misterio, Thriller
Journey's End	2017	Drama, Guerra
The Stoneman Murders	2009	Crimen, Drama, Misterio
Horatio Hornblower 3	2003	Aventura, Drama, Guerra
1971	2007	Acción, Drama, Guerra
The Cruel Sea	1953	Drama, Guerra

TABLA 5.8: Nombre, año y género de las películas más frecuentes en el tópico 7: Guerra.

la trama para tratar de deducir si alguno de los géneros con los que estaban catalogadas era más acorde. De hecho las *word clouds* de los tópicos 0 y 1 que llamamos 'Romance/ Comedia' y 'Drama/Romance', respectivamente, comparten varias palabras como se verá más adelante (ver sección 5.3, o figura 5.3, más precisamente).

NOMBRE	AÑO	GÉNERO
La passione di Giosue l'Ebreo	2005	Drama
Hercules	2005	Aventura, Drama, Fantasía
Arn: Tempelriddaren	2007	Acción, Drama, Romance
Kingdom of Heaven	2005	Acción, Aventura, Drama
The Egyptian	1954	Biografía, Drama, Historia
Ajooba	1991	Acción, Aventura, Comedia
Samson and Delilah	1996	Aventura, Drama, Historia
Samson	2018	Acción, Drama
Arn: Riket vid vagens slut	2008	Acción, Romance, Drama
Sampoorna Ramayanam	1958	Fantasía

TABLA 5.9: Nombre, año y género de las películas más frecuentes en el tópico 8: Historia/Fantasía.

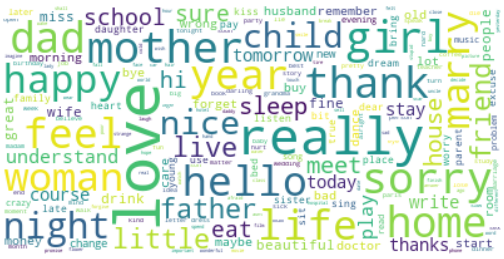
NOMBRE	AÑO	GÉNERO
Cup of My Blood	2005	Horror, Thriller
MVP: Most Vertical Primate	2001	Comedia, Familiar, Deporte
The Nature of the Beast	1995	Crimen, Misterio, Thriller
The Last Light	2014	Drama, Thriller
Insanitarium	2008	Horror, Thriller
Tremors 3: Back to Perfection	2001	Acción, Comedia, Horror
Jack	2014	Drama
Een zaak van leven of dood	1983	Drama, Thriller
Oblivion	2013	Acción, Aventura, Ciencia Ficción
"Beck" Pojken i glaskulan	2002	Crimen, Drama, Misterio

TABLA 5.10: Nombre, año y género de las películas más frecuentes en el tópico 9: Horror/Thriller.

Con las consideraciones mencionadas anteriormente decidimos llamar al tópico 9 "Horror/Thriller", igual que el tópico 6.

5.3. Word clouds

En esta sección incluiremos las *word clouds* generadas a partir de la matriz H de "tópicos por palabras" (10×2.766) del método NMF. Recordamos de la sección 2.1.4 que estas son una forma visual para representar la matriz pues contiene la información del conjunto de términos que componen los tópicos y su importancia en los mismos (peso en la matriz). El peso del término i en el tópico j, es decir el elemento h_{ij} de la matriz H indica el tamaño del token en la *word cloud* correspondiente al tópico j.



(A) Romance/Comedia (B) Drama/Romance

(C) Documental (D) Comedia

(E) Drama (F) Acción/Crimen

(G) Horror/Thriller (H) Guerra

(I) Historia/Fantasía (J) Horror/Thriller

FIGURA 5.3: Word clouds correspondientes a los diez tópicos generados con el método NMF, de la librería sci-kit learn en Python, sobre la base de datos del texto presente en los subtítulos de películas. Los nombres de los tópicos se eligieron en base a las diez películas más representativas de cada tópico. Buscamos en la base de datos de películas en línea, IMDb (https://www.imdb.com/), los géneros de cada una y le asignamos el o los más frecuente/s.

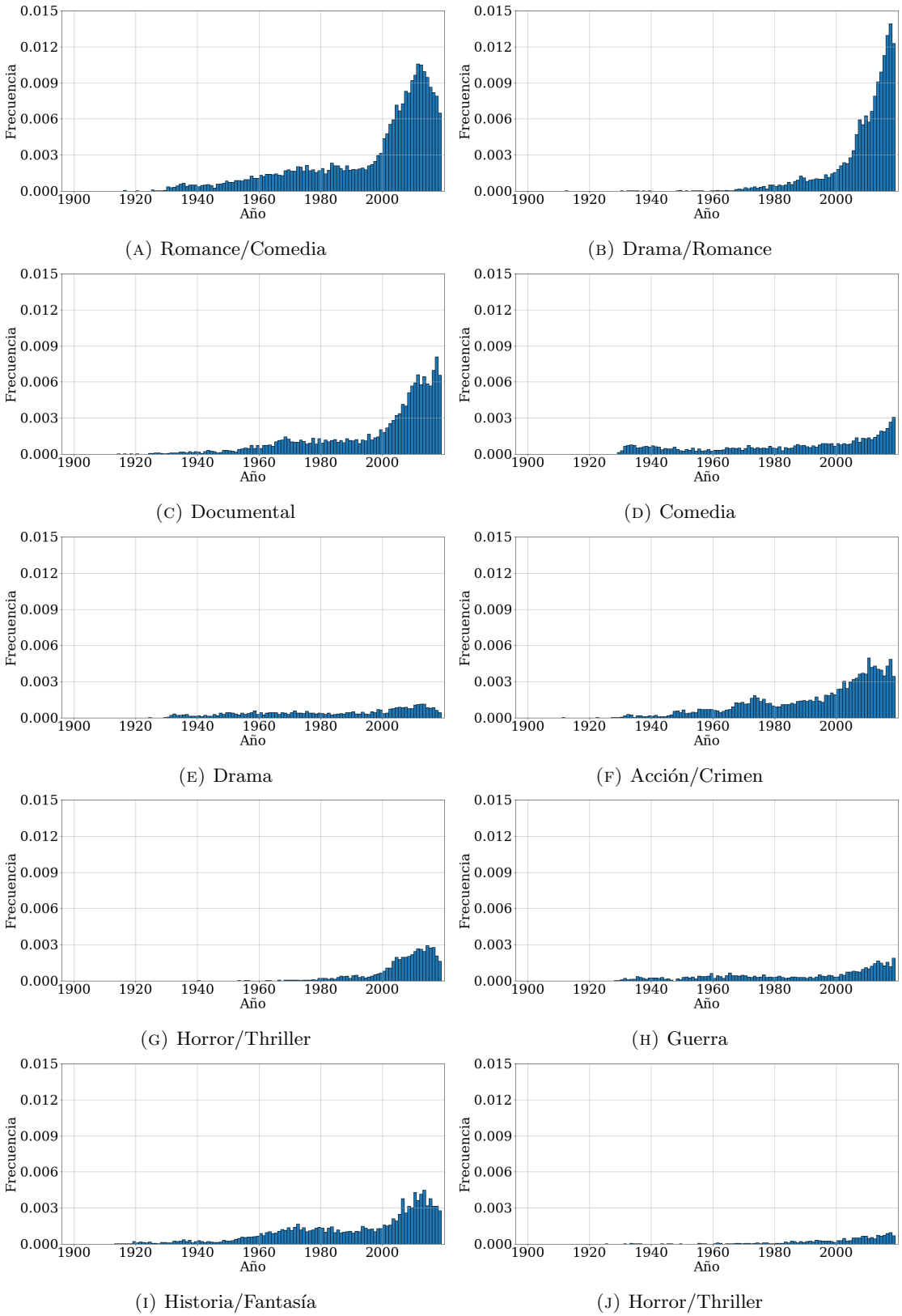
En estas imágenes podemos ver que hay términos específicos en cada tópico importantes para entender su contenido semántico. Por ejemplo en el tópico que llamamos "Romance/Comedia" (5.3a) encontramos las palabras 'love', 'happy', 'life', 'marry' mientras que en el denominado "Documental" (5.3c) vemos 'people', 'war', 'world'. Que un token esté en la *word cloud* de algún tópico no significa que no pueda estar en la de otro. En el caso de estos dos ejemplos encontramos la palabra 'life', que dicho sea de paso tienen una presencia parecida en ambos tópicos según su tamaño. Y aún más, como anticipamos en la sección anterior, los tópicos "Romance/Comedia" y "Drama/Romance" comparten 'life', 'really', 'sorry', 'dad', 'little', etc., en mayor o menor medida. Lo que define entonces a un tópico es el conjunto **todos** de los términos y el peso de cada uno.

Generalmente los tópicos más claros en cuanto a su significado son aquellos que tienen mayor cantidad de palabras únicas o de mayor peso. Se puede ver en el tópico "Acción/ Crimen" las palabras 'money', 'police', 'kill'; o en el tópico "Historia/Fantasía", 'king', 'lord', 'master'; en "Guerra", 'sir', 'captain', 'sergeant'; o las ya mencionadas en el tópico 2 "Documental". Ver figura 5.3.

En la *word cloud* del tópico 9 "Horror /Thriller", figura 5.3j, observamos numerosos nombres propios, tales como 'Jack', 'Mary', 'Charlie', 'Harry', etc. Esto se debe a que no encontramos un método eficiente para eliminar los nombres propios al preprocesar el texto de los subtítulos de las películas. También pensamos que descartando el 5 % de las palabras menos frecuentes a través de la función **TfidfVectorizer()** éstas palabras serían removidas. No obstante, como veremos en la siguiente sección, este tópico es el que menos aporta al volumen total de películas.

5.4. Estadística de los tópicos

La matriz W de "películas por tópicos" (54.731×10) contiene la información del grado de pertenencia de las películas a los 10 tópicos en los que divide la matriz de contenido semántico el algoritmo del método NMF. Si le asignamos a las películas el tópico que más la representa, es decir para la película i buscamos la mayor componente de la fila i de la matriz W, supongamos W_{ij} entonces a la película i la etiquetamos con el tópico j. A continuación mostraremos los histogramas de cantidad de películas por tópicos por año, normalizados por el número total de películas (54.731).



(A) Romance/Comedia

(B) Drama/Romance

(C) Documental

(D) Comedia

(E) Drama

(F) Acción/Crimen

(G) Horror/Thriller

(H) Guerra

(I) Historia/Fantasía

(J) Horror/Thriller

FIGURA 5.4: Histogramas correspondientes a los diez tópicos generados con el método NMF, de la librería sci-kit learn en Python, sobre la base de datos del texto presente en los subtítulos de películas. Los nombres de los tópicos se eligieron en base a las diez películas más representativas de cada tópico. Buscamos en la base de datos de películas en línea, IMDB (https://www.imdb.com/), los géneros de cada una y le asignamos el o los más frecuente/s.

En principio, al comparar los histogramas de la figura 5.4 podemos observar una fuerte diferencia entre la cantidad de películas categorizadas por tópico entre los 10 totales. Tenemos que entonces, nuestra base de datos está compuesta mayormente por películas de "Drama/Romance", seguidas por las de "Romance/Comedia" y "Documentales". Por otro lado, la proporción de películas asociadas a los tópicos "Drama", "Guerra" y "Horror/Thriller" son las menores dentro de las 54.731 totales.

A su vez, podemos notar que en los últimos 20 años se duplicó, triplicó o más en algunos casos, el número de películas publicadas por tópico. Parecería ser que el tópico más antiguo es "Historia/Fantasía" aunque el que tiene más películas entre 1920 y 1940 es el que llamamos "Comedia".

En la última década el último tópico "Horror/Thriller" tiene pocas películas en comparación a los demás. Además, si nos remontamos a su *word cloud* (5.3j) notamos que los tokens más importantes son nombres propios del idioma inglés, los cuales estaban incluidos en el diccionario inglés de NLTK. Incrementar el número de tópicos más allá de 10 resultó en tópicos redundantes o de difícil interpretación, coincidente con el número de tópicos sugerido por el *elbow method*.

Por último, pero no por eso menos importante, queremos destacar que el algoritmo del método **NMF()** no aglomeró en los tópicos a las películas según su año de publicación. Esto que parece una obviedad no lo va a ser cuando discutamos la longitud de los caminos del grafo semántico en la sección 3.3.2.

5.5. Bump charts

Una forma complementaria de visualizar la evolución temporal de tópicos, además de los histogramas, es estudiando la dependencia del peso de los tópicos en función del tiempo (antes de asignarle a cada película un tópico particular). En la figura 5.5 mostramos el bump chart normalizado del peso de los tópicos en función del tiempo. Para confeccionarlo usamos bines de 10 años, es decir que cada 10 años sumamos el peso de los tópicos de todas las películas que se publicaron dentro de dicha ventana temporal (sumamos para cada tópico los valores de su respectiva columna en la matriz W cuyas filas correspondían a películas publicadas en las fechas deseadas). Es un bump chart normalizado ya que para todos los años se ve la fracción que representa el peso de cada tópico respecto del total de películas publicadas en ese tiempo, por más que los primeros años hay una cantidad de películas despreciable en comparación a la totalidad de la base de datos (ver tabla 5.11).

(1889,1899]	(1899,1909]	(1909,1919]	(1919,1929]	(1929,1939]	(1939,1949]	(1949,1959]
1	4	57	169	1175	1236	2022
(1959,1969]	(1969,1979]	(1979,1989]	(1989,1999]	(1999,2009]	(1999,2019]	
3115	4056	4271	5381	13181	20063	

TABLA 5.11: Cantidad de películas por bines de diez años.

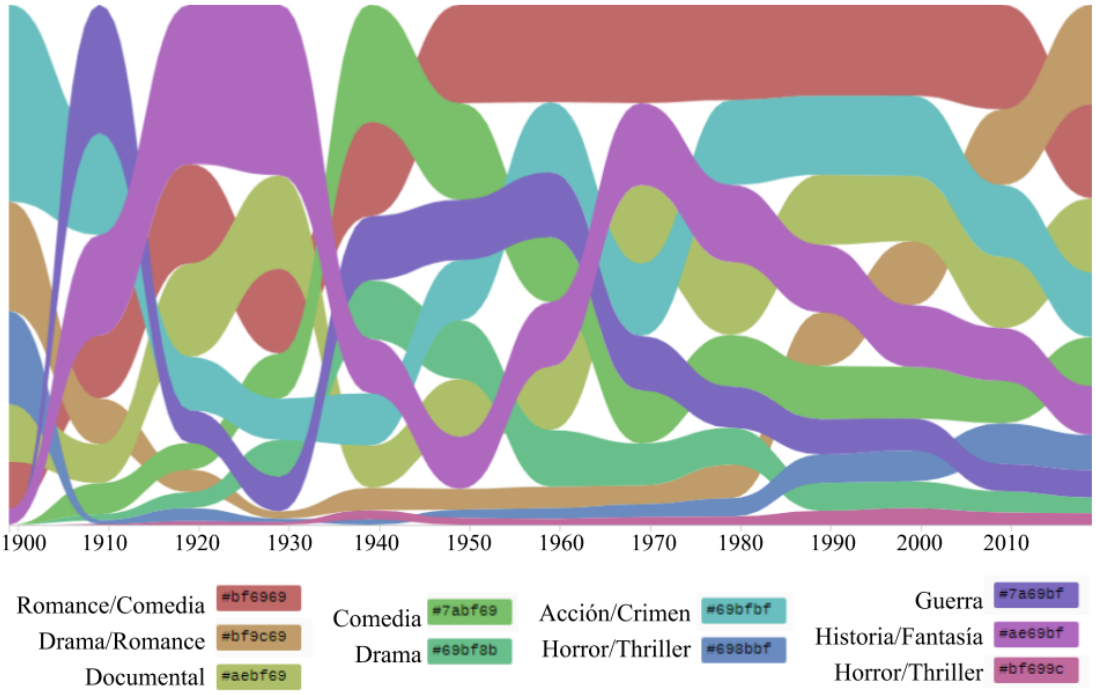


FIGURA 5.5: Bump chart normalizado en bines de 10 años, del peso de los tópicos.

En el gráfico de la figura 5.5 podemos observar un aumento del ranking del tópico "Guerra" en los períodos correspondientes a los dos conflictos bélicos más devastadores de la historia de la humanidad. Una lectura posible a este fenómeno es que para la época de la Segunda Guerra Mundial, muchos estudios pudieron prestar activos servicios creando filmes más o menos propagandísticos de apoyo a los Aliados, y en contra del Eje. También se puede ver que el incremento del ranking del género "Historia/ Fantasía" procede al de "Guerra". Este comportamiento lo podemos intentar adjudicar a dos fenómenos. El primero tiene que ver con la intención de las productoras de desviar la atención de la sociedad de la realidad, de los desastres causados por la guerra. Y el segundo, con que este género es en el que se centraron las grandes productoras para ganarle a la popularidad creciente de la televisión gracias a la potencial espectacularidad propia de la pantalla grande (Bazin, 2012).

El tópico 9 "Horror/Thriller" de color rosa en el bump chart de la figura 5.5 es el más bajo en el ranking y sólo pasa al tópico homónimo en un período.

Realizamos también un bump chart normalizado de la fracción de películas publicadas por tópico en bines de 10 años. Aquí sí le asignamos a cada película el tópico que correspondía a la mayor componente en la fila de la matriz W de "películas por tópicos" de esa película, es decir el tópico que más la representaba. En la figura 5.6 se puede ver el mismo, que tiene la información presente en los histogramas de la sección anterior 5.4 normalizada en los bines de la tabla 5.11. De esta forma, se puede visualizar más claramente la diferencia cronológica entre las proporciones de películas por cada tópico, pudiendo distinguir que en los inicios de la historia fílmica casi la totalidad de películas estaban relacionadas a los géneros de "Romance/Comedia", "Horror/Thriller" e "Historia/Fantasía" además del inicial de "Acción/Crimen". Luego, la variedad de géneros comenzó a ampliarse rápidamente y a partir de la década del 40 comienzan a tener más peso los demás, a pesar de seguir siendo más frecuentes unos que otros. Si observamos los últimos años del mismo gráfico, podemos notar que esta distinción entre la producción de películas de diferente género sigue vigente. De hecho, podemos notar un comportamiento creciente (aparentemente exponencial observando el histograma de la figura 5.4b) para la cantidad de películas de "Drama/Romance" llegando a ser la más frecuente en la actualidad, habiendo partido de los últimos puestos en la década del 70.

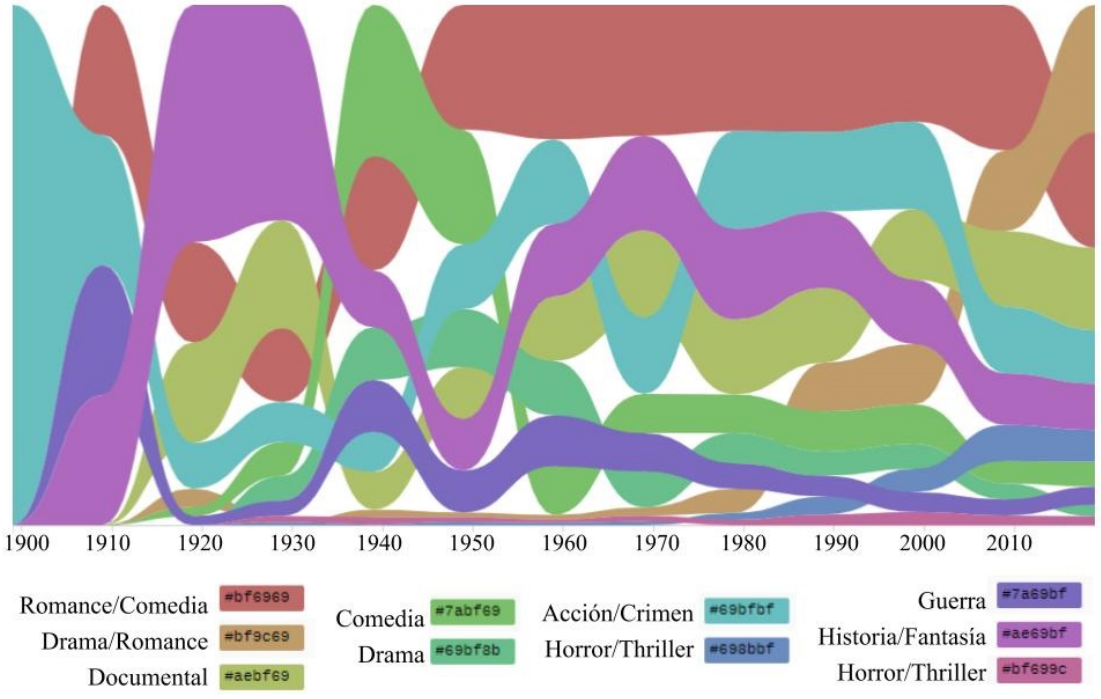


FIGURA 5.6: Bump chart normalizado en bines de 10 años, de películas por tópicos.

En los gráficos de las figuras 5.5 y 5.6 se puede ver que los pesos de los tópicos y la cantidad de películas por tópico se comportan de forma similar, a excepción de los años correspondientes a principio de siglo. Esto es producto de tener pocas películas en los primeros bines. En particular se puede observar que para el primer bin hay una sola

película (tabla 5.11), que fue categorizada como "Acción/Crimen" (figura 5.6) debido a que este tópico es el de mayor peso (figura 5.5). Lo que quiere decir que si bien categorizamos las películas con un sólo tópico, también contienen en su desarrollo semántico elementos no despreciables de otros tópicos. Para el segundo bin, tenemos cuatro películas y por lo tanto cuatro géneros para ese bin en la figura 5.6 pero una fracción apreciable de siete tópicos en la figura 5.5. Ya a partir del tercer bin ambos gráficos se asemejan notablemente.

5.6. Visualización en 2D con t-SNE

La matriz W de "películas por tópicos" (54.731×10) tiene 54.731 vectores de dimensión 10 representados en un espacio semántico. A través de la función **TSNE()** de la biblioteca *sci-kit learn* en Python convertimos estos vectores de dimensión 10 en *embeddings* de 2D. Los *embeddings* pueden ser visualizados fácilmente mediante puntos en un espacio de 2D. Cada punto a su vez puede pensarse como los nodos de un grafo. La distancia entre nodos viene dada por la similitud entre los embeddings, cuanto más similares los vectores se grafican más cerca el uno del otro, y cuanto menos parecidos, más lejos. Esta forma de representación se puede observar en la figura 5.7. Allí se muestra la evolución temporal del grafo construido con la matriz de adyacencia semántica. Vemos los clústers correspondientes a los 10 tópicos del grafo que contiene las películas (nodos) publicadas **hasta** la fecha volcada en la figura. El color de los nodos indica la pertenencia a determinado clúster (tópico) calculada mediante el peso máximo del vector de dimensión 10 de la película.

En esta figura notamos que el clúster con más películas hasta 1944 era el correspondiente al género "Comedia" pero que a partir de allí casi ni aumentó su volumen con respecto a los demás clústers. También se puede ver que el clúster que más aumentó en el último tramo, del 2004 al 2019 es el de "Drama/Romance" que es consistente con los histogramas mostrados anteriormente, figura 5.4. El clúster más pequeño es el que representa al tópico 9 "Horror/Thriller".

Si observamos el último grafo de la figura, *publicadas hasta 2019*, notamos una intersección de tópicos importante entre los nodos azules, rojos y naranjas (correspondientes a los tópicos "Romance/Comedia, Comedia y Drama/Romance, respectivamente). Esto se debe a que el contenido semántico de los vectores de esas películas son similares puesto que los *embeddings* producidos por la función **TSNE()** lo son. Si recordamos la sección 5.2, donde elegimos los nombres de los tópicos, estos géneros fueron los que más debate

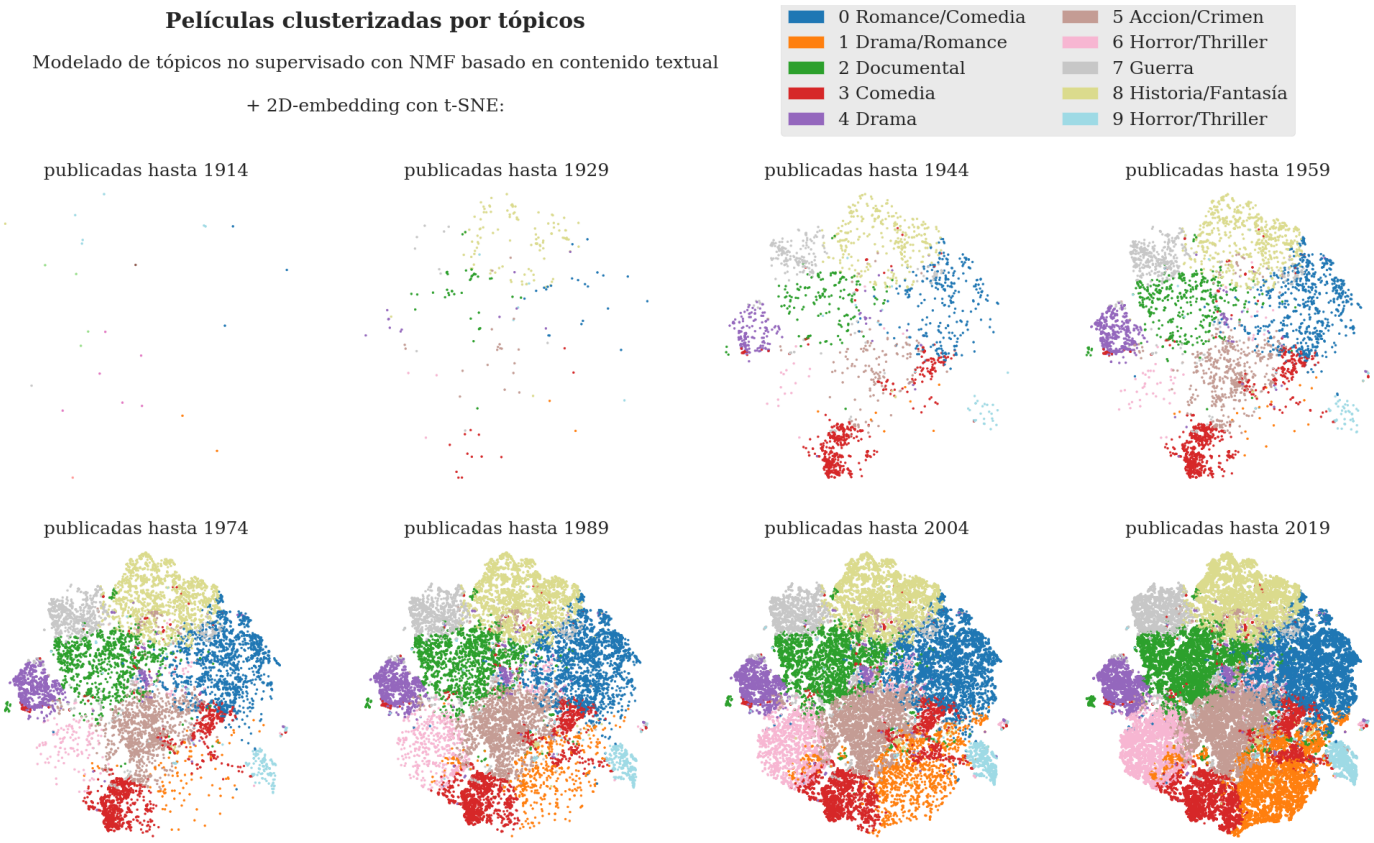


FIGURA 5.7: Visualización en 2D con t-SNE de la cantidad de películas por clústers (tópicos) a lo largo del tiempo. En cada gráfico se muestra el tamaño de los clústers de la red hasta ese momento.

generaron porque estaban repetidos en muchas de las 10 películas más representativas de cada tópico.

También vemos que los tópicos verde y gris se encuentran juntos en el grafo. Si volvemos a la tabla de género de las 10 películas más representativas del tópico 2 "Documental" podemos notar que hay varias películas que además de 'documental' tienen el género 'guerra' o el nombre de la misma indica su temática.

Una última observación es que el clúster turquesa (tópico 9 "Horror/Thriller") está bastante alejado de su homónimo rosa. En un principio, al debatir los nombres de los tópicos, pensamos que podía ser un sólo género que fue dividido en dos por el algoritmo de NMF, lo cual es posible. Además está cerca del clúster azul "Romance/Comedia" y si vemos la tabla 5.10 nos encontramos con "MVP: Most Vertical Primate" una película familiar de 'comedia' como segunda película más representativa de ese tópico. Además, encontramos allí a "Tromens 3" una película de 'horror' y 'comedia'. Por lo que pudimos concluir que, al contrario de nuestra suposición, no eran el mismo género que fue dividido en dos tópicos distintos.

Capítulo 6

Similitud a lo largo del tiempo

En este capítulo discutiremos el método que utilizamos para estimar la optimalidad de la distancia cronológica en el grafo de contenido semántico de nuestra base de datos, introducida previamente en la sección 3.3.2.

6.1. Construcción del grafo semántico

Como ya dijimos en los capítulos de métodos, extrajimos las líneas de subtítulos de los archivos *.srt* de las películas de la base de datos de opensubtitles y los volcamos en un dataframe de *Pandas* en Python. Luego, utilizamos un filtro de duración y otro de cantidad de palabras de los subtítulos de las películas. Con esto "limpiamos" los datos lo máximo posible y obtuvimos el dataframe filtrado con las líneas de subtítulos, intervalos temporales donde aparece cada línea en pantalla, y el nombre y año de las 54.731 películas filtradas.

Utilizando la función **TfidfVectorizer()** construimos la matriz semántica en representación TF-IDF del contenido de las líneas de subtítulos de las películas. Y con la función **NMF()** la descompusimos en las matrices W de "películas por tópicos" y H de "tópicos por palabras" dándole como parámetro 10 en el 'número de tópicos'. Al calcular el coeficiente de correlación de Pearson para los vectores de películas de la matriz W obtuvimos la matriz de correlación del contenido semántico de los subtítulos de películas. Y es esta última matriz de correlación la matriz de adyacencia A con la que construimos el grafo semántico. Este grafo es pesado, no dirigido y se puede decir que está completamente conectado.

6.2. *Ensembles* de caminos

La matriz de adyacencia del grafo semántico contiene los pesos de los enlaces entre nodos. Este peso al ser calculado mediante la correlación de Pearson indica la similitud semántica entre los nodos, ya que esta era la información contenida en los vectores de película con el que fue calculado. Es decir, si dos vectores de películas son similares (tienen contenido semántico similar ya que los pesos correspondientes a los tópicos son parecidos) la correlación entre ellos va a ser cercana a 1. Si son opuestos, será cercana a -1, y si son totalmente diferentes va a tender a 0.

En este sentido, si queremos recorrer el grafo semántico pasando por todos sus nodos y sumando el peso de los enlaces por los que nos movemos, no va a ser lo mismo ir del nodo 1 al nodo 2 o del nodo 1 al nodo 3 si su contenido semántico es distinto. Supongamos que todos los nodos representan la misma película, entonces vamos a tener 54.731 nodos con idéntica información y el peso de la matriz de adyacencia será 1 puesto que la correlación de Pearson entre dos vectores perfectamente co-lineales da 1. En este caso si recorremos todo el grafo sumando los pesos de los enlaces por los que pasamos la cuenta dará exactamente 54.730 ya que es la cantidad de enlaces por los que tengo que transitar para llegar a 54.731 nodos. Aunque hagamos alguna permutación en el camino que recorremos, si seguimos recorriendo todo el grafo, la cuenta seguirá dando lo mismo.

Si ahora cambiamos uno de los nodos por una película con contenido semántico opuesto al de los otros 54.730 nodos, los enlaces de ese nodo tendrán peso -1 y ya no dará lo mismo la cuenta si permutamos el camino. Esto se debe a que si el *nodo opositor* se encuentra en uno de los extremos (el inicio o el final del camino) sólo habremos sumado uno de sus enlaces y la cuenta dará 54.728. En cambio si el *nodo opositor* se encuentra en el medio del camino, cualquier permutación sumará 54.726, ya que habremos sumado dos de sus enlaces.

Otro caso útil para entender la cuenta es si tenemos 10 nodos, de los cuales 5 corresponden a una película A y los otros 5 a otra con contenido semántico opuesto, la película B. Ahora los nodos iguales tendrán enlaces con peso 1 y los nodos opuestos, peso -1. Nos preguntamos: ¿En qué caso sumaremos lo máximo posible? La respuesta será en el caso que recorramos el grafo priorizando la similitud semántica entre los nodos. Es decir, si empezamos a recorrerlo por un nodo tipo A, deberíamos ir a otro del mismo tipo para sumar 1 y así sucesivamente hasta que no haya más nodos del tipo A. En ese caso saltamos a un nodo tipo B, sumamos -1 y luego nos dirigimos a otro nodo tipo B que sumamos 1 otra vez. De este modo sumaríamos 7. La cuenta se va haciendo más compleja a medida que agregamos nodos con contenido semántico distinto y permutamos el orden de los nodos.

Siguiendo este principio (que el camino más pesado es el que prioriza la similitud del contenido semántico) realizamos la cuenta que habíamos introducido en la sección 3.3.2 para estudiar si hay una evolución cronológica del contenido semántico de los subtítulos de películas a lo largo del tiempo. Para ello, recorrimos el grafo semántico de cuatro formas distintas pero siempre manteniendo el inicio y el fin del camino.

La primer forma de recorrerlo fue la cronológica. Ordenamos las películas por año en el dataframe filtrado y construimos una lista L cuyos elementos son listas que contienen los índices de las películas de determinado año, para cada año, y ordenadas cronológicamente. Luego generamos otra lista, I, que concatena los elementos de L, es decir, las listas de años con el índice de las películas de ese año. Como la matriz de adyacencia tiene los mismos índices para sus nodos que el dataframe filtrado, hicimos una función que suma el peso del enlace entre dos nodos consecutivos de la lista I, es decir la componente $a_{I_i,I_{i+1}}$ de la matriz A para $i = 0, ..., N-1$ con $N = 54.731$. A esta suma, que corresponde a la longitud del camino, la llamamos "Longitud de Camino Ordenado".

Luego construimos un iterador de 10.000 ciclos[1] que en cada uno de ellos hace una permutación aleatoria de los elementos en las listas de años de L, manteniendo el primer y último elemento de I, vuelve a generar I y computa la longitud de este nuevo "camino ordenado". También hace una permutación aleatoria de todos los índices de películas dentro de I, salvo el primero y último, como se muestra en la figura 3.2 de la sección 3.3.2, y calcula la longitud de este "camino aleatorio". De esta manera por cada ciclo obtenemos un valor para la longitud del camino ordenado cronológicamente (pero cambiando el orden de los nodos del mismo año) y otro valor para la del camino aleatorio (que en principio no respeta orden alguno).

En la figura 6.1 podemos ver los histogramas para estos dos tipos de caminos, el ordenado cronológicamente y el aleatorio. Dibujamos sobre los 10.000 datos de cada camino una función gaussiana centrada en el promedio y con la desviación estándar de los mismos.

Podemos ver que el promedio del *ensemble* de caminos ordenados es mayor al *ensemble* de caminos aleatorios por aproximadamente el 20 %. Con esto podemos inferir que el contenido semántico de los subtítulos de películas es similar para películas contemporáneas a sus épocas, sin importar el género de las mismas.

Hicimos el mismo estudio pero el camino ordenado ahora lo construimos con los índices de las películas ordenando el dataframe filtrado por año y luego por tópicos (luego de haberle asociado el tópico que más la representaba). Es decir que para cada año tenemos

[1] Dado que tenemos 54.731 nodos, hay 54.729! formas de recorrer los mismos permutándolos manteniendo los extremos fijos. Este número es exageradamente grande por lo que decidimos usar un número más razonable en tiempo de cómputo para el iterador.

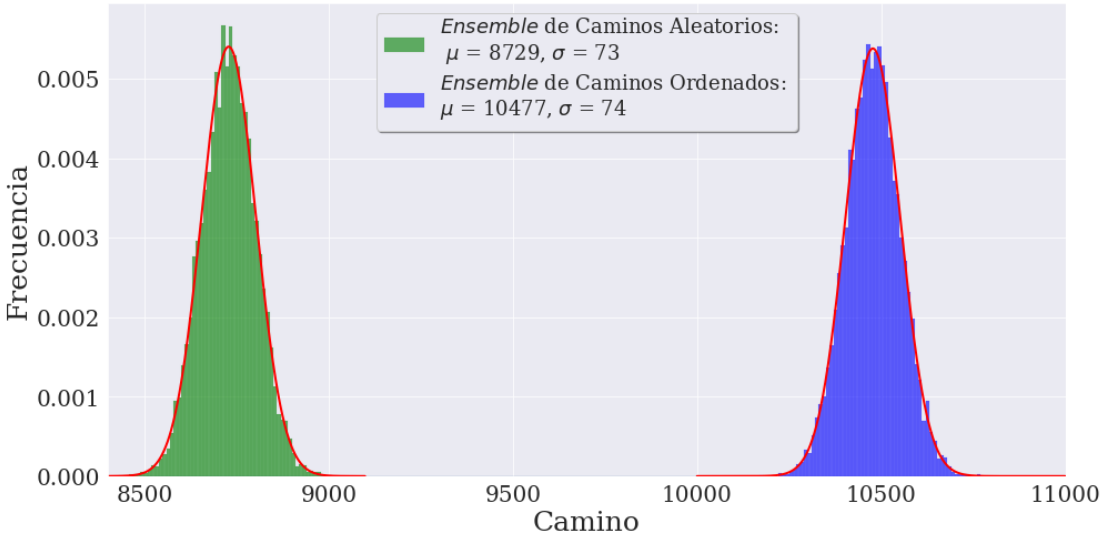


FIGURA 6.1: Histogramas para los caminos ordenado por años y aleatorio. En rojo se grafica una Gaussiana $N(\mu, \sigma)$ para cada histograma. Notamos una diferencia al comparar la distribución de caminos aleatorios y ordenados, siendo superador el orden cronológico.

una lista conteniendo 10 listas (una por cada tópico) con los índices de todas las películas de un mismo tópico. Al igual que en el caso anterior utilizamos el iterador pero esta vez las permutaciones del camino ordenado eran sobre los índices dentro de cada una de las 10 listas de tópico, por año. El resultado se puede ver en la figura 6.2.

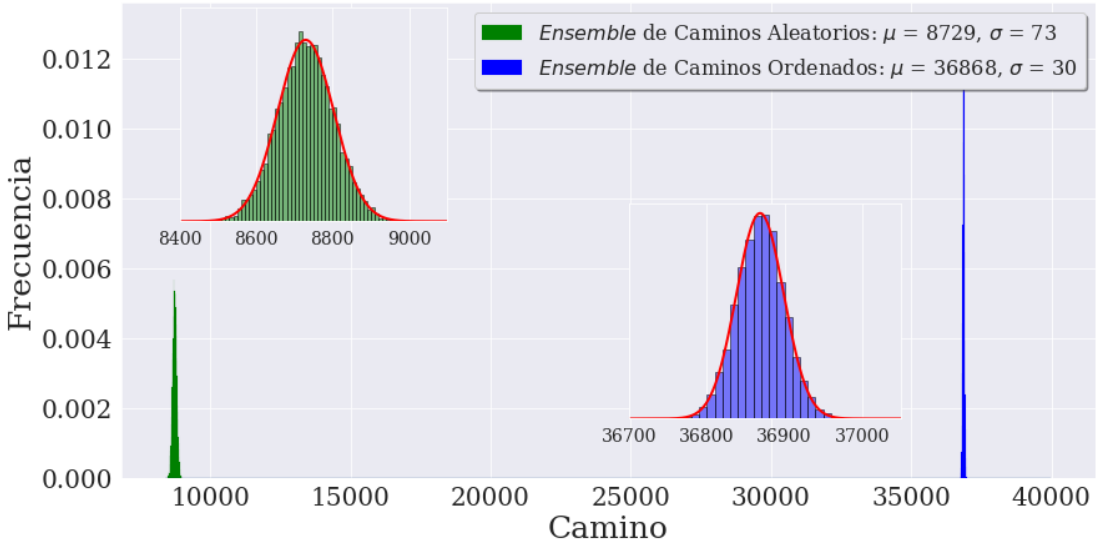


FIGURA 6.2: Histogramas para los caminos ordenado por años y luego por tópico, y aleatorio. Se muestra un zoom de cada histograma y en rojo se grafica una Gaussiana $N(\mu, \sigma)$ para cada uno. Al comparar con el orden anterior, observamos que si recorremos los caminos no sólo cronológicamente, sino luego por tópico, aumenta significativamente el promedio de la distribución de estos caminos.

Como se puede ver, el promedio del *ensemble* de caminos ordenados por año y tópico,

es más de cuatro veces mayor que el promedio del *ensemble* de caminos aleatorios y más de tres veces el promedio del *ensemble* de caminos ordenados sólo por año. Con esto podemos concluir que el contenido semántico en el texto de los subtítulos de películas es muy similar entre películas de un mismo tópico.

Repetimos el mismo procedimiento pero ahora ordenando el dataframe filtrado sólo por tópicos, es decir todas las películas del tópico 0 al principio, luego las del 1, y así sucesivamente. Las permutaciones del camino ordenado fueron entre los índices de las películas de cada tópico sin importar el año de las películas. En la figura 6.3 mostramos los histogramas para los *ensembles* de este caso.

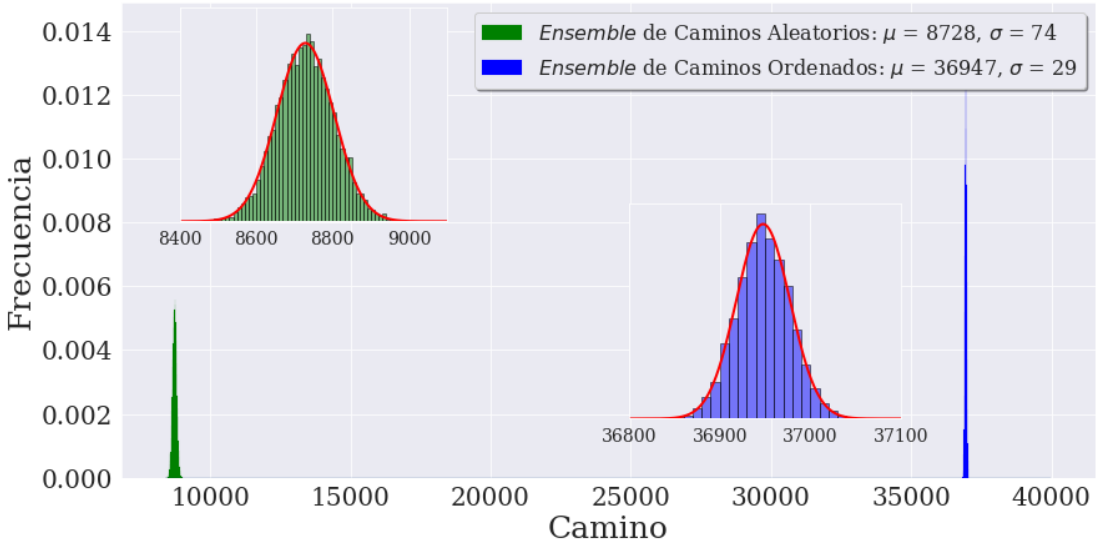


FIGURA 6.3: Histogramas para los caminos ordenado por tópico, y aleatorio. Se muestra un zoom de cada histograma y en rojo se grafica una Gaussiana $\mathrm{N}(\mu, \sigma)$ para cada uno. Podemos notar también en este caso que al ordenar su similitud semántica, se obtiene un camino mayor que si fuera primero por años y luego por tópico.

Al igual que en los casos anteriores, la distribución de caminos aleatorios fue casi idéntica mientras que el camino ordenado sigue siendo un orden de magnitud mayor al promedio y prioriza recorrer el grafo por el contenido semántico de las películas en los tópicos.

Por último, ordenamos el dataframe filtrado con la combinación que faltaba, por tópicos y luego por año. Se puede ver en el gráfico de la 6.4 que obtuvimos el valor de camino ordenado más grande. Esto nos motiva a pensar que la similitud semántica de los subtítulos de películas trasciende el orden por tópicos y podría estar relacionada con fenómenos sociales producidos en distintas épocas.

Calculamos la función peso para el grafo semántico. Para ello computamos la suma de cada fila de la matriz adyacente A y lo graficamos en el histograma que se puede ver en la figura 6.5. Como se observa a simple vista, es una distribución no homogénea que

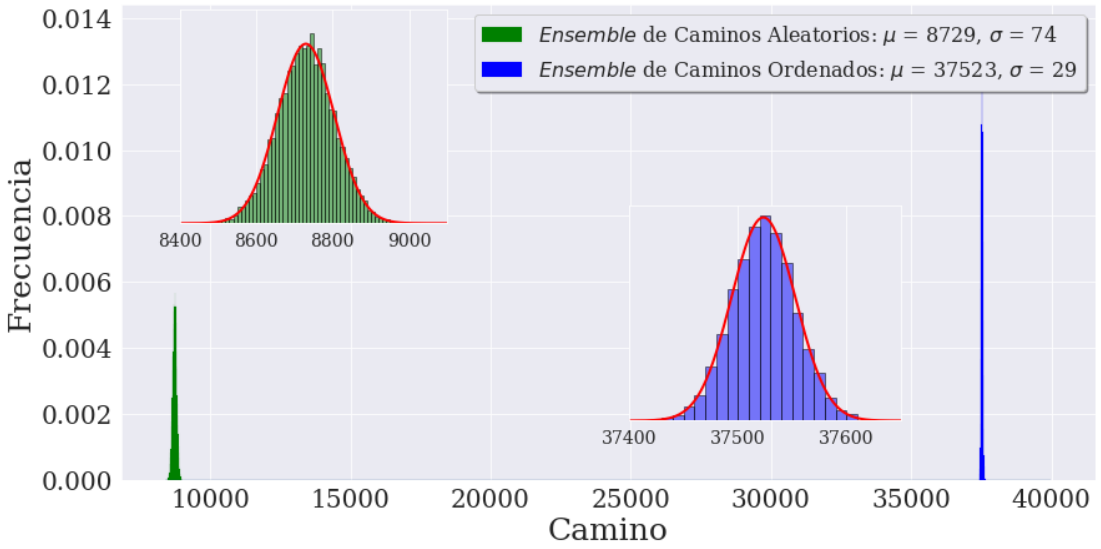


FIGURA 6.4: Histogramas para los caminos ordenado por tópicos y luego por años, y aleatorio. Se muestra un zoom de cada histograma y en rojo se grafica una Gaussiana $N(\mu, \sigma)$ para cada uno. A través de este orden, obtuvimos el mayor valor para el promedio de la distribución de caminos.

se mueve aproximadamente entre los valores de -15.000 hasta 20.000. El promedio de los valores de la suma de pesos por fila es de 8.730, lo resulta consistente con los valores obtenidos al recorrer el grafo de manera aleatoria, pasando por todos los nodos.

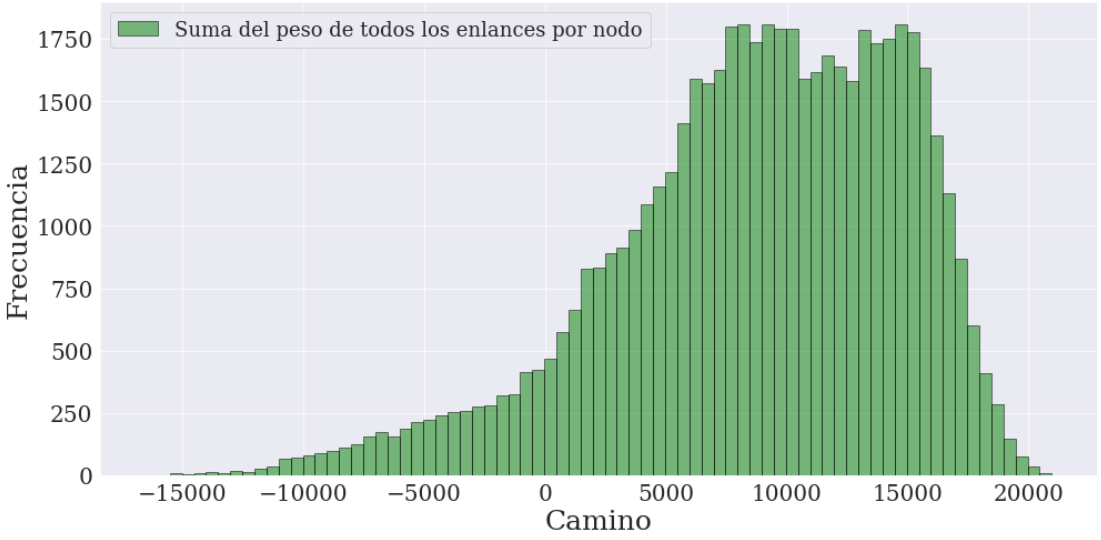


FIGURA 6.5: Histograma correspondiente a la distribución del peso de todos los enlaces por nodo. Al sumar estos pesos observamos que esta distribución no es homogénea y al calcular el promedio se obtiene un valor de 8.730, consistente con los valores obtenidos al recorrer el grafo sin ningún orden específico.

Todas las figuras que mostramos en esta sección nos muestran que el contenido semántico de los subtítulos de películas fue evolucionando a lo largo del tiempo manteniendo una similitud notoria para películas con fecha de publicación cercanas en el tiempo independientemente del género de las mismas.

Capítulo 7

Conclusiones

En esta tesis se estudió una extensa base de datos de subtítulos provenientes de opensubtitles.org. Realizando un filtrado y un procesado de esta base de datos, nos quedamos con un total de 54.731 subtítulos de películas, las cuales caracterizamos mediante histogramas de duración y cantidad de palabras. La motivación de un análisis con este tipo de datos es poder encontrar la respuesta a la pregunta de cómo determinar numéricamente si hay o no presente un cambio consistente para una período determinado de tiempo tal que las obras de una época se pueden considerar similares entre sí, a pesar incluso de tratarse de contenidos semánticos diferentes.

Para esto, el primer paso fue encontrar un número de tópicos semánticos apropiado para distinguir al total de subtítulos mediante el método NMF. Con esto en mente se estudió el gráfico del error de reconstrucción del método y se detectó un pequeño pero significativo cambio en la derivada de la función error para 10 tópicos semánticos al descomponer la matriz, en representación TF-IDF, del texto de los subtítulos de películas en las matrices de menor dimensión W "documentos por tópicos" y H "tópicos por palabras". Inspeccionando manualmente los tópicos obtenidos y su contenido, corroboramos que 10 tópicos fueron capaces de reproducir acertadamente las temáticas generales de las películas.

Aplicamos entonces el método NMF para detectar estos 10 tópicos y asociarlos al género de la película. Se tomó un criterio para esta asociación: tomar las 10 películas más representativas de cada tópico, esto es, las diez primeras que tuvieran el mayor valor en la columna j de la matriz W para el tópico j, y luego buscar en otra base de datos online, IMDb, el o los género/s correspondiente/s a esas películas. De esta forma, optamos por los géneros más frecuentes de las 10 películas elegidas para cada tópico y se obtuvieron los siguientes: "Romance/Comedia", "Drama/Romance", "Documental", "Comedia", "Drama", "Acción/Crimen", "Horror/Thriller", "Guerra", "Historia/

Fantasía", y nuevamente "Horror/Thriller". Si bien este criterio fue el elegido para los análisis posteriores, se podría estudiar con mayor profundidad si es el más apropiado en un análisis a futuro. Dado que no es lo principal en este trabajo, consideramos que es lo suficientemente adecuado para lo que queremos ilustrar.

Es importante notar que si bien existen trabajos que utilizan bases de datos de subtítulos para evaluar la performance de distintos análisis de detección de tópicos (Rabinovich and Girdhar, 2015; Mocanu et al., 2016; Bougiatiotis and Giannakopoulos, 2017), no conocemos ningún trabajo que evalúe la evolución temporal de los mismos en una base de datos grande como la que utilizamos aquí, en el espíritu de otros trabajos que investigan patrones de evolución cultural en libros (Michel et al., 2011), cuadros (Kim et al., 2014) o música (Levitin et al., 2012). Un objetivo futuro será aplicar métodos similares a otras formas de expresión artística con el objetivo de investigar su co-evolución temporal y su mutua interdependencia.

Para visualizar qué palabras son las que más influyen en determinar el nombre de cada tópico, confeccionamos las word clouds para cada uno de ellos. Éstas muestran de manera gráfica la frecuencia de una palabra en cada tópico (tamaño de la palabra en la nube). Al buscar correspondencias entre lo que aparece en cada word cloud y el tópico que le fue asociado, encontramos que los tópicos que más discusión generaron para la elección de su nombre compartían varias palabras. Además los que más fáciles de determinar resultaron, tenían palabras únicas y de mayor peso que las de los demás. Por ejemplo, se obtuvo que para el tópico "Acción/Crimen" las palabras más frecuentes en orden eran: 'money', 'police', 'kill'.

Por otro lado, asignamos a cada película el tópico correspondiente a la columna de la matriz W con el mayor peso de la fila de la película. Al estudiar la distribución temporal de las películas por tópico, encontramos que nuestra base de datos está compuesta mayormente por películas "Romance/Comedia", "Drama/Romance" y "Documental", mientras que las presentes que corresponderían a las de "Drama", "Horror/Thriller" y "Guerra" se encuentran en menor proporción. También es importante destacar que no parece haber una aglomeración aparente para la separación de los tópicos según su año de publicación. Esto es un resultado positivo ya que reforzará las conclusiones del cómputo sobre los caminos del grafo semántico.

Se confeccionaron bump charts normalizados para representar la evolución temporal de los tópicos en dos formas: según el peso de los tópicos y según la cantidad de películas por tópico, distribuidas en bines de 10 años. Al analizar esta forma de visualización, pudimos distinguir que a partir de mitad de siglo, donde hay un punto de inflexión en la cantidad de datos por año, los comportamientos de ambos gráficos son similares. Esto nos indica una buena consistencia en que tanto las películas como el contenido

semántico asociadas a un tópico evolucionaron de formas semejantes a partir de esta época. Previamente, no distinguimos este resultado, pero tampoco podríamos concluir que sucede lo contrario ya que no contamos con una cantidad considerable de películas como para poder hacerlo.

También podemos correlacionar el hecho de que haya un aumento de la proporción del tópico "Guerra" en los períodos correspondientes a la Primer y Segunda Guerra Mundial con el desarrollo del mercado del cine. Sabemos que en estos momentos, muchos estudios estuvieron activamente prestando servicios para la producción de películas más o menos propagandísticas en apoyo a los Aliados y en contra del Eje (Fox, 2007).

Al realizar un análisis con la herramienta t-SNE, se puedo apreciar una gran separación entre las películas de diferentes tópicos, mostrando consistencia en la elección de los 10 seleccionados. Si bien esto fue positivo, al observar en detalle las películas correspondientes al tópico: "Comedia" pudimos detectar discrepancias en un clúster al estar separado espacialmente de todo el conjunto. A pesar de estar aglomerados en su mayoría, hay una porción no despreciable que se encuentra entre los clústers asociados a los tópicos de "Romance/Comedia" y "Drama/Romance". Esto nos podría estar indicando que esta asociación puede no ser la adecuada para estas películas o que esta porción corresponde a la intersección de cada tópico, siendo películas "cómicas-romanticonas". Por otro lado, comparando los dos tópicos de "Horror/Thriller" apreciamos notablemente la distancia que los separa, reforzando la idea de que esta etiqueta puede no ser la adecuada, o que hay películas de este género cuyo contenido semántico difiere considerablemente.

Se estudió la matriz de adyacencia al grafo semántico con el objetivo de encontrar alguna correlación entre la forma de recorrer la matriz (para determinar la longitud del camino) y la longitud del camino obtenida a través de esa forma. Se busca de este modo calcular numéricamente si hay una mayor correlación semántica de las películas al recorrer el grafo cronológicamente. Para ello se realizaron 10.000 iteraciones permutando el camino recorrido en el "modo aleatorio", dejando fijo los extremos, y para los siguientes órdenes de los nodos (las películas) del camino: por año y permutando nodos al azar dentro de cada año; por año, luego por tópico y permutando aleatoriamente nodos dentro de cada tópico; por tópico y luego permutando nodos al azar dentro de cada tópico; y finalmente por tópicos, luego por año y después permutando de forma aleatoria nodos dentro de cada año. Se obtuvo que el orden que maximiza la longitud del camino recorrido es al priorizar en principio los tópicos y luego cronológicamente con una suma de 37.523 ± 29. Al ordenar sólo por tópicos, 36.947 ± 29; por año y por tópico, 36.868 ± 30; y por años, 10.477 ± 74. En particular, notamos un incremento de la longitud del camino ordenado respecto del estimado mediante sampleo aleatorio, 8.729 ± 74, al ordenar los nodos únicamente por la fecha de publicación de la película correspondiente. Esto nos

indica que hay presente una afinidad por épocas en el contenido semántico de subtítulos de películas, tanto dentro de los mismos tópicos como independientemente de los mismos.

Como un plus a este último análisis, para estudiar los enlaces de cada nodo se calculó la suma de los pesos de los mismos y se graficó en un histograma la distribución de estas sumas. Se obtuvo una disposición no homogénea llegando aproximadamente a valores de entre -15.000 y 20.000, con un promedio en 8.730, igual al promedio de la longitud de camino al recorrer los nodos del grafo de forma aleatoria.

7.1. Limitaciones y perspectivas futuras

En este estudio nos encontramos con ciertas limitaciones a la hora de analizar la enorme base de datos de subtítulos de películas relacionadas su confección. Al ser generada por humanos, existen errores de tipeo, asignación errónea del nombre o año de la película y el idioma de los subtítulos, entre otros. Sin embargo al contar con una gran cantidad de archivos filtrados, 54.731, supusimos que estos errores no iban a alterar de forma apreciable el análisis de los datos. A su vez, resultó imposible analizar cada subtítulo manualmente para reducir este potencial ruido en los resultados.

Como consideraciones a futuro, encontramos posible hacer un estudio de tokens específicos y visualizar su frecuencia a lo largo del tiempo en el corpus del texto de subtítulos de películas. Esto es lo que muestra el algoritmo de Google, Ngram Viewer (`https://books.google.com/ngrams`) pero para distintos corpus de texto basado en libros. Además podría compararse el resultado del Ngram Viewer y estudiar si hay una correlación entre la frecuencia de aparición del token en los libros y la frecuencia en las películas. Esto sería una aproximación a estudiar cómo distintas formas de expresión artística y cultural pueden influenciarse mutuamente entre sí.

Mediante el algoritmo word2vec se puede entrenar un modelo de vectores de palabras en el corpus para obtener *embeddings* de palabras en un espacio vectorial y por lo tanto plausible para realizar operaciones matemáticas entre los *embeddings*. Usando el etiquetado temporal de las palabras, al provenir de subtítulos, se puede estudiar la evolución temporal ficcional del promedio de la distancia semántica entre ciertos tokens y todas las palabras de una línea subtítulos. Con esto analizar la presencia de modos normales de esta función dentro de los tópicos y compararlos con las seis formas básicas que dominan los arcos narrativos *emocionales* de las historias (Reagan et al., 2016).

Al ver las tablas de la sección 5.2 llama la atención que en algunos tópicos las películas más representativas tengan una fecha de publicación similar. Se puede, entonces, estudiar la *novedad* del contenido semántico de las películas. Con esto encontrar las películas que sentaron las bases del contenido semántico para los distintos tópicos. Es decir buscar las películas con contenido semántico diferente a la de todas sus antecesoras pero similar al de las películas posteriores en fecha de publicación.

Bibliografía

Alghamdi, R. and K. Alfalqi (2015). A survey of topic modeling in text mining. *Int. J. Adv. Comput. Sci. Appl.(IJACSA)*@(1).

Barabási, A.-L. et al. (2016). *Network science*. Cambridge university press.

Barth, L., S. I. Fabrikant, S. G. Kobourov, A. Lubiw, M. Nöllenburg, Y. Okamoto, S. Pupyrev, C. Squarcella, T. Ueckerdt, and A. Wolff (2014). Semantic word cloud representations: Hardness and approximation algorithms. *LATIN 2014: Theoretical Informatics. Lecture Notes in Computer Science 8392, 514–525.*

Bazin, A. (2012). ¿Qué es el cine? *Libros de cine Rialp.*

Bougiatiotis, K. and T. Giannakopoulos (2017). Multimodal content representation and similarity ranking of movies. arXiv preprint arXiv:1702.04815.

Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa (2011). Natural language processing (almost) from scratch. Journal of machine learning research*@(Aug), 2493–2537.*

Evergreen, S. D. (2019). Effective data visualization: The right chart for the right data. *Sage Publications.*

Fox, J. (2007). Film propaganda in Britain and Nazi Germany: World War II cinema. *Berg.*

Goldberg, Y. (2016). A primer on neural network models for natural language processing. Journal of Artificial Intelligence Research 57, 345–420.

Kim, D., S.-W. Son, and H. Jeong (2014). Large-scale quantitative analysis of painting arts. *Scientific reports 4, 7370.*

Lee, D. D. and H. S. Seung (2001). Algorithms for non-negative matrix factorization. In Advances in neural information processing systems*, pp. 556–562.*

Levitin, D. J., P. Chordia, and V. Menon (2012). Musical rhythm spectra from bach to joplin obey a 1/f power law. Proceedings of the National Academy of Sciences*@(10), 3716–3720.*

Loper, E. and S. Bird (2002). Nltk: the natural language toolkit. arXiv preprint cs/0205028*.*

Maaten, L. v. d. and G. Hinton (2008). Visualizing data using t-sne. Journal of machine learning research*@(Nov), 2579–2605.*

McKinney, W. and P. Team (2015). pandas: powerful python data analysis toolkit. Pandas—Powerful Python Data Analysis Toolkit*, 1625.*

Michel, J.-B., Y. K. Shen, A. P. Aiden, A. Veres, M. K. Gray, J. P. Pickett, D. Hoiberg, D. Clancy, P. Norvig, J. Orwant, et al. (2011). Quantitative analysis of culture using millions of digitized books. science*@(6014), 176–182.*

Mocanu, B., R. Tapu, and E. Tapu (2016). Video retrieval using relevant topics extraction from movie subtitles. In 2016 12th IEEE International Symposium on Electronics and Telecommunications (ISETC)*, pp. 327–330. IEEE.*

Pinto, S., F. Albanese, C. O. Dorso, and P. Balenzuela (2019). Quantifying time-dependent media agenda and public opinion by topic modeling. Physica A: Statistical Mechanics and its Applications 524, 614–624.

Rabinovich, M. and Y. Girdhar (2015). Gaining insight into films via topic modeling & visualization. *Parsons Journal of Information Mapping*@(1), 3–5.

Ramos, J. et al. (2003). Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, Volume 242, pp. 133–142. Piscataway, NJ.

Reagan, A. J., L. Mitchell, D. Kiley, C. M. Danforth, and P. S. Dodds (2016). The emotional arcs of stories are dominated by six basic shapes. *EPJDataScience*@(31).

Schmid, H. (1995). Treetagger-a language independent part-of-speech tagger. *Institut fur Maschinelle Sprachverarbeitung, Universitat Stuttgart*.

Sporns, O., G. Tononi, and R. Kötter (2005). The human connectome: A structural description of the human brain. *PLoS Computional Biology*@(4).

Thorndike, R. L. (1953). Who belongs in the family. In *Psychometrika*. Citeseer.

Van Der Aalst, W. (2016). *Data science in action*. Springer, Berlin, Heidelberg.

Xu, W., X. Liu, and Y. Gong (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 267–273.

Buenos Aires, 2020