Práctica Final
Predicción del género de libros

Inteligencia Artificial en las Organizaciones

Grupo 83-1

Miguel Gutiérrez Pérez
100383537@alumnos.uc3m.es

Mario Lozano Cortés
100383511@alumnos.uc3m.es

Alba Reinders Sánchez
100383444@alumnos.uc3m.es

Alejandro Valverde Mahou
100383383@alumnos.uc3m.es

GitHub: ***InteligenciaArtificialOrganizaciones***

10 de diciembre de 2020

Índice

1. Introducción

El objetivo de esta práctica consiste en abordar una clasificación sobre resúmenes de libros para determinar su género literario. Las razones que llevan a la elección de este problema tienen que ver con que actualmente cualquier persona con la dedicación suficiente puede escribir un libro sin la necesidad del patrocinio de una editorial, lo que conlleva una **explosión en el número de nuevos libros generados**. Por consiguiente, las librerías y bibliotecas necesitan catalogar una gran cantidad de escritos, lo cual, les lleva a necesitar de métodos de clasificación automática. Por ello, se plantea el uso de resúmenes y metadatos de los libros puesto que la tarea de clasificación **debe poder realizarse con el menor número de datos posible**, puesto que no todos los libros que llegan a estas entidades disponen de todos los datos completos.

La primera cuestión imprescindible que surge al conocer el problema propuesto es qué técnica de Inteligencia Artificial emplear. Dado que se realiza un análisis sobre diferentes textos, la opción evidente es la **Minería de Texto**, la cual es una técnica de minería de datos que busca extraer **información útil y relevante de documentos de texto** de diferentes fuentes diferentes, como puede ser páginas web, correos electrónicos, periódicos o redes sociales. Para ello, se hace una identificación de patrones en los datos, como puede ser la repetición de palabras o conjuntos de palabras, estructuras sintácticas que se repitan a lo largo de los datos, etc. Esta minería de texto tiene numerosas aplicaciones, y en esta práctica se van a desarrollar una clasificación en función de unas categorías que serán definidas gracias a la elección de un dataset apropiado.

A continuación se ofrece un **esquema del funcionamiento de la tarea propuesta** en donde un libro sin catalogación llega a alguna a de estas entidades que necesitan catalogar su género a partir de la información más reducida posible (generalmente título y argumento). Inicialmente se plantea la distinción de un único género, sin embargo, **es bien sabido que un escrito no tiene por qué adscribirse a un único género** y por lo tanto se debe considerar como futura **ampliación** catalogar tantos como sea posible.

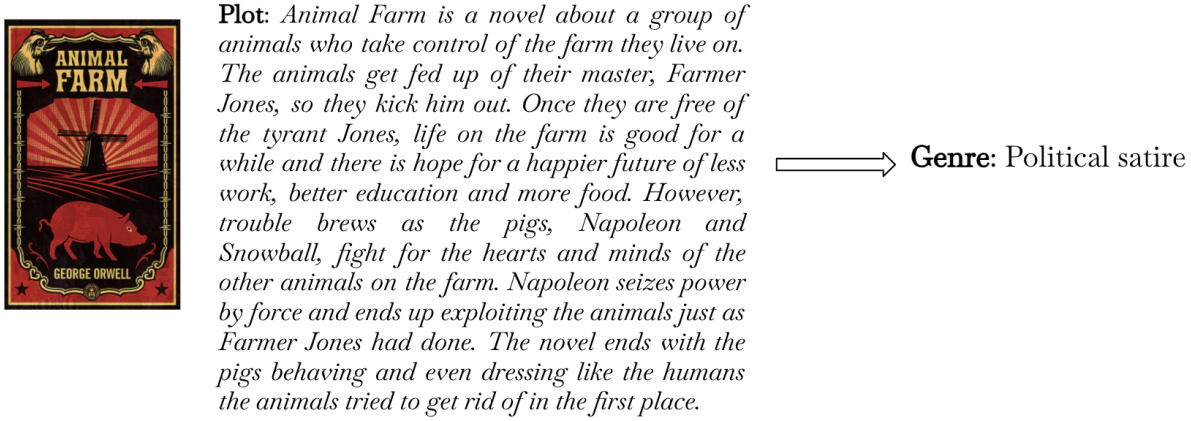

Figura 1: Esquema de la tarea

2. Conjunto de datos

En esta sección se describirá el **conjunto de datos** utilizado en su forma original, así como todos los cambios que se vayan a hacer como parte del **preprocesado** junto con las razones que han llevado a su realización.

2.1. Estructura original

El conjunto de datos ha sido obtenido de **Kaggle** [INSERTAR], y dispone de un total de **54283 libros y 12 columna** con información sobre ellos. En la Figura X se puede apreciar visualmente la forma de las instancias de este conjunto de datos.

Author(s)	Description	Edition	Format	ISBN	No. Pages	Avg. Rating	No. Ratings	No. Reviews	Title	Genres	Cover Image
William Golding	At the dawn of the next world war, a plane crashes on an uncharted island, stranding a group of...	Penguin Great Books of the 20th Century	Paperback	9.78E+12	182	3.66	1840595	30634	Lord of the Flies	Classics|Fiction|Young Adult|Academic|School|Literature	Link

Figura 2: Instancia del conjunto de datos original

A continuación se va a explicar el **contenido** de cada columna en mas detalle:

- **Author(s):** Cadena de caracteres que indica el o los autores del libro. En caso de ser mas de uno, cada autor aparece separado por '|'.
- **Description:** Cadena de caracteres que indica el resumen o descripción del libro.
- **Edition:** Cadena de caracteres que indica la edición del libro.
- **Format:** Cadena de caracteres que indica el formato del libro, como por ejemplo *handcover* o *paperback*.
- **ISBN:** Valor numérico que indica el ISBN del libro. Debido a que se utiliza notación científica para su representación, no se puede leer ni usar correctamente, pues faltan números.
- **No. Pages:** Valor numérico que indica el número de páginas que tiene el libro.
- **Avg. Rating:** Valor numérico que indica la valoración media que los usuarios han dado al libro.
- **No. Ratings:** Valor numérico que indica al cantidad de valoraciones de usuario que ha recibido el libro.
- **No. Reviews:** Valor numérico que indica la cantidad de críticas que ha recibido el libro.
- **Title:** Cadena de caracteres que indica el título del libro.
- **Genres:** Cadena de caracteres que indica los géneros a los que pertenece el libro. Los diferentes géneros aparecen separados por '|'.
- **Cover Image:** Imagen que muestra la portada del libro. Se indica un enlace que lleva hasta dicha imagen, aunque por motivos de espacio no se ha incluido en la figura de arriba.

Inicialmente se consideró usar el conjunto de datos **CMU Book Summary Dataset** [INSERTAR], pero la manera en la que estaban dispuestos los géneros, así como el resto de atributos, daba lugar a un **procesado más complejo** para obtener sus valores. Además, se consideró

que los géneros que se utilizaban no eran muy acertados, como por ejemplo *Speculative fiction* o *Postmodernism*. Posteriormente se encontró el conjunto de datos que se ha descrito mas arriba, el cual tenía muchas mas instancias (54,283 frente a 16,559), proporcionaba mucha más información (tenía mas atributos), y, lo mas importante, era mucho mas sencillo obtener los valores de sus instancias.

2.2. Preprocesado

En toda tarea de minería de texto existe algún tipo de **preprocesado**, ya sea obtener solo los valores que se necesiten, eliminar instancias con errores, vectorizar el texto, etc. Esta vez no va a ser diferente, e incluso se puede afirmar que ha sido una **parte importante del trabajo**, y que ha ocupado una cantidad considerable de tiempo. A continuación se detallan todas las **modificaciones** que se han llevado a cabo:

- **Selección de las columnas útiles:** De las 12 columnas que tiene el conjunto de datos seleccionado, solo son necesarias dos, *description* y *genres*, que contenían, respectivamente, el resumen o descripción del libro y los géneros que se le atribuyen. Por ello, solo solo se mantendrán dichas columnas eliminando el resto.
- **Eliminación de instancias con descripciones en un lenguaje diferente al ingles:** Los resúmenes del conjunto de datos se encuentran escritos no solo en inglés, si no también en árabe, italiano, chino, coreano, japonés, portugués...Se ha decidido por razones evidentes, mantener solo aquellos libros cuyos resúmenes estén en ingles. Para la detección del idioma se ha utilizado la librería ***langdetect*** [https://pypi.org/project/langdetect/] de *Python*, y aquellas instancias donde se detectaba un idioma diferente al inglés se han borrado. Gracias a la gran cantidad de instancias que hay, la eliminación de las instancias no afecta de manera significativa.
- **Corrección de formatos incorrectos:** Algunos de los resúmenes del conjunto de datos contenían errores de formato como saltos de línea (de diferentes tipos), tabulaciones o espacios en medio de los resúmenes, así que se sustituyeron por un único espacio en blanco.
- **Eliminación de caracteres no útiles:** Se han eliminado de los resúmenes del conjunto de datos caracteres que no proporcionaban ningún tipo de información, como comillas, guiones, corchetes, paréntesis, puntos, exclamaciones, interrogaciones, etc.
- **Eliminación de términos no útiles:** Se han eliminado de los resúmenes del conjunto de datos términos que no proporcionaban ningún tipo de información útil. Para ello se ha utilizado una lista de ***stopwords***, reutilizada de la segunda práctica de la asignatura, pero a la cual se han añadido términos que se han ido encontrado al realizar pruebas. Evidentemente, no se han incluido todos los términos sin información útil posibles, pero dada la gran cantidad de términos que puede haber, creemos que la cantidad de ***stopwords*** obtenida es suficiente.
- **Cambios requeridos por la aproximación usada:** La aproximación que se utilizada en este trabajo requiere que los resúmenes esté en minúsculas. Al contrario que ocurría en la segunda práctica, no es necesario dividir los resúmenes en términos, por lo que no se utilizara la cadena de caracteres que contiene el resumen de manera directa.
- **Eliminación de géneros no útiles:** Después de analizar un poco más a fondo los diferentes géneros que se usan, se ha observado la falta de un criterio claro para elegir que géneros usar, dando lugar a géneros de países, como *spain*, al mismo tiempo que el género *spanish literature*. También se encontraron géneros muy concretos y sin mucho sentido, como *Amazon* o *Apple*. Esto llevó a que fuera necesario revisar manualmente todos los

géneros, los cuales eran en torno a 850, y eliminar aquello que se consideraron no útiles. Al finalizar esta revisión quedaron 625 géneros.

Una vez aplicado todo esto, los datos ya están preparados para su uso. Sin embargo, queda una última cosa. La cadena de caracteres que contiene los géneros se separara según el caracter '|' en los diferentes géneros para su posterior uso. Como en este trabajo se va explorar una clasificación **multi-clase** con el primer género, y una clasificación ***multi-label*** con todos los géneros, para la primera solo se utilizará el primer género, y para la segunda se usarán todos.

3. Conceptos teóricos

A la hora de elegir un método de Minería de Textos se consideran diversas opciones tales como utilizar la herramienta *Weka* de la *Universidad de Waitako*, sin embargo, dado que esta herramienta ya ha sido utilizada a lo largo de la asignatura a la que se adscribe este trabajo y además presenta ciertas limitaciones en cuanto a la flexibilidad y capacidad de toma de decisiones de diseño en los modelos se decide aplicar **Word Embedding** en redes de neuronas con la **biblioteca de código abierto Tensorflow**. De esta manera se usará un método diferente al usado en clase, lo cual permite experimentar y aprender tecnologías nuevas.

No obstante, antes de iniciar la codificación de algún modelo, dado que se trata de una técnica nueva es necesario tener claros los conceptos teóricos involucrados. Una red neuronal únicamente procesa números, lo que implica que es necesario realizar una transformación. Representar las palabras como vectores es importante, ya que los modelos de inteligencia artificial no 'entienden' las palabras, y no puede realizar cálculos ni aprendizaje sobre ellas. Por este motivo, es necesario realizar una **vectorización**, que no es más que transformar estas palabras en números, agrupados en forma de vector. A continuación se exponen las diferentes técnicas de vectorización consideradas.

3.1. Codificación con valor único

Una primera aproximación podría ser asignar un único número a cada una de las palabras consideradas en la vectorización. Así si por ejemplo, se tiene la frase "hace un espléndido día"se obtiene un vector como el que sigue:

Palabra	Valor
hace	1
un	2
espléndido	3
día	4

Tabla 1: Codificación con un único valor

Este enfoque presenta una serie de consideraciones importantes:

- El valor de cada palabra se decide de manera arbitraria.
- No se obtiene una representación fiel de la distancia entre palabras, lo cual es un hecho que es importante en el desarrollo de esta práctica.

Por ello, se descarta el enfoque aquí propuesto por no ser eficiente en la tarea propuesta.

3.2. One-hot encoding

Al codificación one-hot consiste en convertir cada palabra en un vector con tantas posiciones como palabras tengamos, 1 en la posición que se corresponda a la palabra considerada y 0 en caso contrario.

A continuación se muestra un ejemplo de la codificación con el pequeño set de palabras utilizado en la sección anterior.

	hace	**un**	**espléndido**	**día**
hace	1	0	0	0
un	0	1	0	0
espléndido	0	0	1	0
día	0	0	0	1

Tabla 2: One-hot encoding

El principal problema de esta técnica de vectorización es que la distancia entre las palabras es la misma, lo cual impide de nuevo disponer de una representación realista de este hecho esencial en el enfoque que se propone.

3.3. Embedding

Está técnica sirve para representar aquellas **palabras que son semánticamente parecidas con una codificación similar**. Lo que hace esta técnica tan atractiva es que no es necesario especificar esta similitud de forma manual. Un *embedding* es un vector denso de números reales, donde su longitud viene determinada por parámetro. En lugar de determinar los pesos a mano, se tratan como **parámetros que pueden ser entrenados**, como si de pesos de redes de neuronas densas se trataran. Por eso mismo, esta técnica funciona especialmente bien con las redes de neuronas, que incorpora la modificación de estos pesos a la fase de entrenamiento de la red.

La dimensionalidad del *embedding* debe ser proporcional a la cantidad de datos disponibles, ya que cuanto más grande sea el vector, más detalles podrá obtener de cada palabra, pero requiere de más datos para ser entrenado. A continuación se muestra un ejemplo de la codificación con el pequeño set de palabras utilizado en la sección anterior.

hace	2.32	7.35
un	0.89	4.23
espléndido	0.75	8.65
día	2.65	3.00

Tabla 3: Embedding

Por lo tanto esta técnica resulta de gran utilidad a la hora de conseguir representar la distancia semántica entre las palabras que forman un texto. Siendo este hecho fundamental para la tarea se propuesta **se decide apostar por esta codificación para construir el modelo de text mining**.

4. Clasificación

Normalmente un libro se encuadra en más de un género literario, por ello, de manera inicial se decide enfocar el problema en la clasificación de un solo género para posteriormente, poder realizar una ampliación del algoritmo que prediga todos los géneros posibles de un escrito.

4.1. Arquitectura del modelo

Una vez se cargan los datos preprocesados se realiza una aleatorización de los mismos para evitar posibles sesgos, se vectorizan las salidas y se dividen los datos en conjunto de entrenamiento (70 %) y en conjunto de test (30 %). Los modelos considerados siguen todos la misma estructura de Red de Neuronas:

- **Capa de vectorización del texto**: Transforma cadenas de caracteres a índices de vocabulario, el vocabulario se crea a partir de la frecuencia de valores individuales. Se modifican los siguientes parámetros:
  - **Número máximo de tokens**: Representa el tamaño del vocabulario a usar.
  - **Longitud de la secuencia de salida**: número de índices de palabras que se pasa a la capa siguiente.
- **Capa de *embedding***: A partir del vocabulario busca el vector de *embedding* para cada índice de palabras. Estos vectores se aprenden según el modelo se entrena.
- **Capa de agrupación promedio global**: Devuelve un vector de salida de longitud fija para cada ejemplo haciendo la media sobre la dimensión de la secuencia. Se utiliza para permitir a la red usar los datos del *embedding*.
- **Capa densa**: Neuronas completamente conectadas. Según el modelo puede tener una o varias capas y el número de neuronas puede variar por capas.
- **Capa de salida**: Capa densa que tiene tantas neuronas como géneros distintos haya.

De forma gráfica se puede presentar la estructura como se muestra en la figura 2.

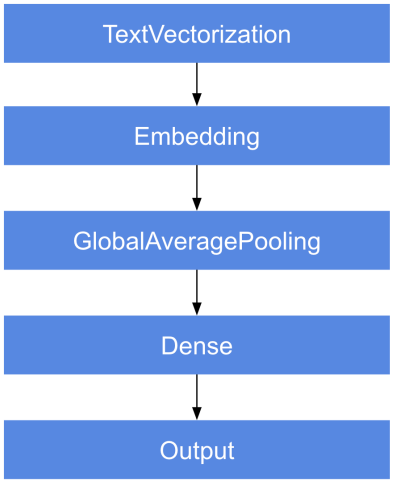

Figura 3: Arquitectura del modelo

4.2. Clasificación de 1 género

La idea de usar un sólo género a la hora de clasificar un libro es útil si este es género es el principal del mismo, sin embargo, el dataset original no hace una diferenciación sobre esto y establece todos los géneros de un libro con la misma importancia. Por ello, se ha decidido guardar el primer género de cada libro, aunque esto **podría suponer un sesgo** sobre el modelo que se debe tener en cuenta. Por ejemplo, si un libro pertenece a los géneros Ciencia ficción, Aventura y Fantasía, al entrenar únicamente con el primer género puede que la salida la red lo clasifique como Fantasía y lo trate como un error al desconocer que también pertenece a este género. Por ello, es de suma importancia realizar la ampliación multigénero.

Al realizar las transformaciones oportunas seleccionando el primer género **se obtienen un total de 194 géneros diferentes**. Se trata por lo tanto de un problema de clasificación multiclase ya que se tienen 194 clases y cada instancia tiene una sola etiqueta.

Para poder ser utilizadas por la red, las salidas se vectorizan usando *one-hot encoding* para representar a estas clases, donde cada posición es un género distinto. Un 0 significa que no pertenece a ese género y un 1 que sí pertenece. Por lo tanto, la forma que tiene el vector de una salida cualquiera es: [0, 0, . . . , 0, 1, . . . , 0]. El cual tiene tamaño 194 y un solo 1.

Otro aspecto importante que hay que tener en cuenta al tratarse de un problema multiclase es la función de coste. Se usa **entropía cruzada categórica** (*Categorical Cross-Entropy*) porque se desea entrenar a la red para sacar como salida la probabilidad para todas las clases sobre cada uno de los ejemplos.

La figura 3 muestra un ejemplo de la codificación one-hot en la vectorización de los géneros de la novela 1984.

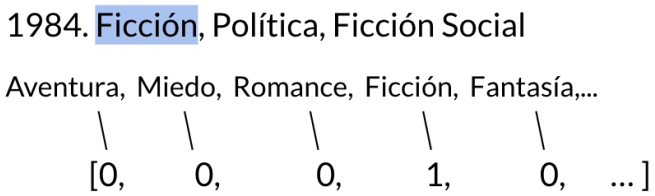

Figura 4: One-hot encoding de 1 género

5. Conclusiones