



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	2
2. Conceptos teóricos	3
2.1. Codificación con valor único	3
2.2. One-hot encoding	3
2.3. Embedding	4
3. Clasificación	5
3.1. Arquitectura del modelo	5
4. Conclusiones	6

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 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.



Title: Animal Farm

Plot: *Animal Farm is a novel about a group of animals who take control of the farm they live on. The animals get fed up of their master, Farmer Jones, so they kick him out. Once they are free of the tyrant Jones, life on the farm is good for a while and there is hope for a happier future of less work, better education and more food. However, trouble brews as the pigs, Napoleon and Snowball, fight for the hearts and minds of the other animals on the farm. Napoleon seizes power by force and ends up exploiting the animals just as Farmer Jones had done. The novel ends with the pigs behaving and even dressing like the humans the animals tried to get rid of in the first place.*

⇒ **Genre:** Political satire

Figura 1: Esquema de la tarea

2. 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.

2.1. 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.

2.2. 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.

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.

	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

2.3. Embedding

Esta 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**.

3. 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.

3.1. 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.

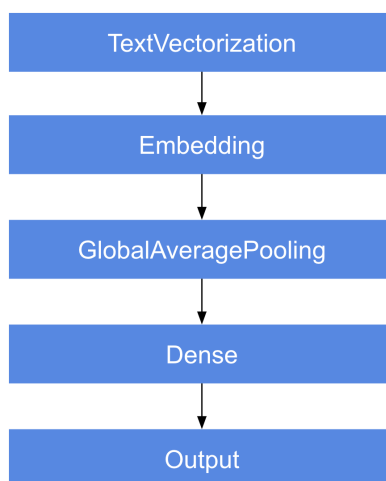


Figura 2: Arquitectura del modelo

3.2. 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: $[0, 0, \dots, 0, 1, \dots, 0]$. 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.

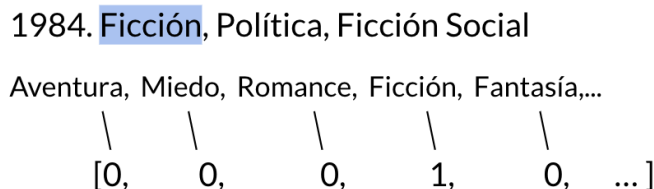


Figura 3: One-hot encoding de 1 género

4. Conclusiones